MICROHOBBY SEMANAL

REVISTA INDEPENDIENTE PARA USUARIOS DE ORDENADORES SINCLAIR

AÑO II- N.º 12

95 PTAS.



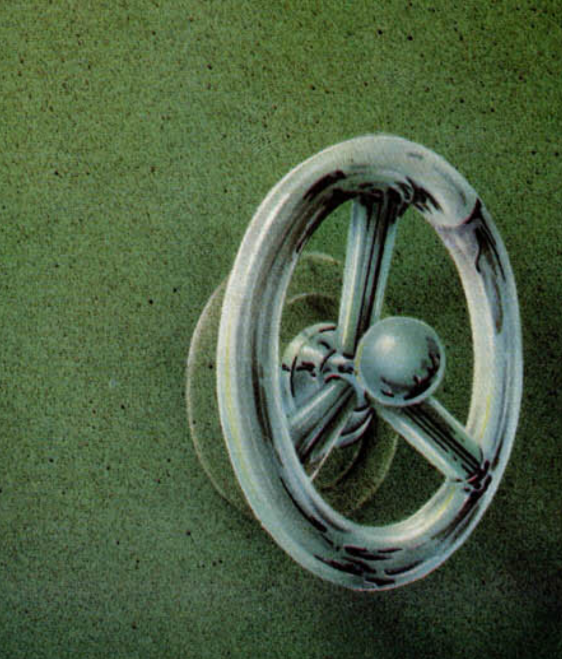
Canarias 105 ptas.

PROGRAMAS

- QUINIELAS: ¡CONSIGUE UNA DE CATORCE!
- LAS TORRES DE HANOI
- ¡ES LA GUERRA!

CURSO BASIC

COMANDOS DE CONTROL DEL SPECTRUM



YO PROTEJO, TU PIRATEAS, EL COPIA



PROGRAMADORES

LOS AUTORES DEL "FRED": UN EQUIPO ESPAÑOL EN LAS LISTAS INGLESAS

Ponce

15.000 PTAS. POR TU PROGRAMA

Director Editorial
José I. Gómez-Centurión
Director Ejecutivo
Domingo Gómez
Redactor Jefe
Africa Pérez Tolosa
Diseño
Jesús Iniesta
Maqueta
Rosa M. Capitel
Redacción
José María Díaz
Gabriel Nieto
Colaboradores
Jesús Alonso, Lorenzo Cebeira, Primitivo de Francisco, Rafael Prades, Víctor Prieto
Fotografía
Javier Martínez
Carlos Candel
Portada
José María Ponce
Dibujos
Manuel Berrocal, J.R. Ballesteros, A. Perera, F.L. Frontán, J. Septien, J.M. López Moreno
Edita
HOBBY PRESS, S.A.
Presidente
María Andrino
Consejero Delegado
José I. Gómez-Centurión
Administrador General
Ernesto Marco
Jefe de Publicidad
Marisa Esteban
Secretaria de Publicidad
Concha Gutiérrez
Publicidad Barcelona
Isidro Iglesias
Tel.: (93) 307 11 13
Secretaria de Dirección
Marisa Cogorro
Suscripciones
M.ª Rosa González
M.ª del Mar Calzada
Redacción, Administración y Publicidad
La Granja, n.º 8
Polígono Industrial de Alcobendas
Telf.: 654 32 11
Distribución
Coedis, S.A. Valencia, 245. Barcelona.
Imprime
Rotedic, S.A.
Carretera de Irún, Km. 12,450
Tel.: 734 15 00
Fotocomposición
Consulgraf
Nicolás Morales, 34 - 1.º
Tel.: 471 29 08
Fotomecánica
Zescán
Nicolás Morales, 38
Tel.: 472 38 58
Depósito Legal:
M-36.598-1984
Representante para Argentina, Chile, Uruguay y Paraguay, Cía. Americana de Ediciones, S.R.L. Sud América, 1.532. Telf.: 21 24 64. 1209 BUENOS AIRES (Argentina).
Derechos Exclusivos
«Sinclair Users», «Sinclair Programs» y «Sinclair Projects» de EMAP Publications (Londres).

Se solicitará control OJD

MICROHOBBY ESTA SEMANA

Año II · N.º 12 · 22 al 28 de enero de 1985
95 ptas. (Sobretasa Canarias 10 ptas.)

OFERTA EXCEPCIONAL DE SUSCRIPCION, VALIDA SOLO HASTA EL 30 DE ENERO DE 1985

MICROHOBBY SEMANAL AHORA A SU ALCANCE ¡¡lleno de ventajas!!



AHORRE 850 PTAS. SOBRE EL PRECIO REGULAR DE SUSCRIPCION ¡¡UN 18% DE DESCUENTO!!



PRECIO PARA VD.
3.900 PTAS.

AHORRO 850 PTAS.



CONSIGA UN REGALO SEGURO. **Gratis** para usted **una de estas tres cintas** de programas, cuyo precio en la calle es de 2.000 PTAS.

¡ELIJA LA QUE QUIERA!



3

PARTICIPE EN VALIOSOS SORTEOS. Cada mes, durante el período de validez de esta oferta, sortearemos entre todos los cupones de suscripción recibidos UN ORDENADOR QL Y TRES MICRODRIVES CON SU INTERFACE:

4 premios valorados en más de 260.000 PTAS.

¡¡CUANTO ANTES RESPONDA MAYORES SERAN SUS OPORTUNIDADES DE GANAR!!





ASEGURESE HOY EL RECIBIR, SEMANA TRAS SEMANA DURANTE TODO UN AÑO, MICROHOBBY: LA REVISTA MAS INNOVADORA Y AGIL EN EL MUNDO DEL SPECTRUM. (50 NUMEROS AL AÑO).



5

DEVUELVANOS SU TARJETA DE SUCRIPCION AHORRO HOY MISMO Y PARTICIPE YA EN EL TERCER SORTEO QUE TENDRA LUGAR ANTE NOTARIO DURANTE LA SEGUNDA SEMANA DE FEBRERO DE 1985

6

PARA CUALQUIER CONSULTA, LLAMENOS A LOS TELS.: 733 50 12
733 50 16
O ESCRIBANOS A HOBBY PRESS, S.A.
C/ Arzobispo Morcillo, 24.
Of. 4. 28029 MADRID.

SI LO DESEA, SOLICITE SU SUSCRIPCION POR TELEFONO.

TRUCOS

ALMACENAR PROGRAMAS CON EL MISMO NOMBRE

Una de las limitaciones del sistema operativo de los microdrives es que no permite la existencia, en un mismo cartucho, de dos programas con el mismo nombre; la forma de evitar esto y conseguir «esquivar» al sistema operativo, es la siguiente: basta con introducir en el nombre del programa los códigos de color directamente; por ejemplo:

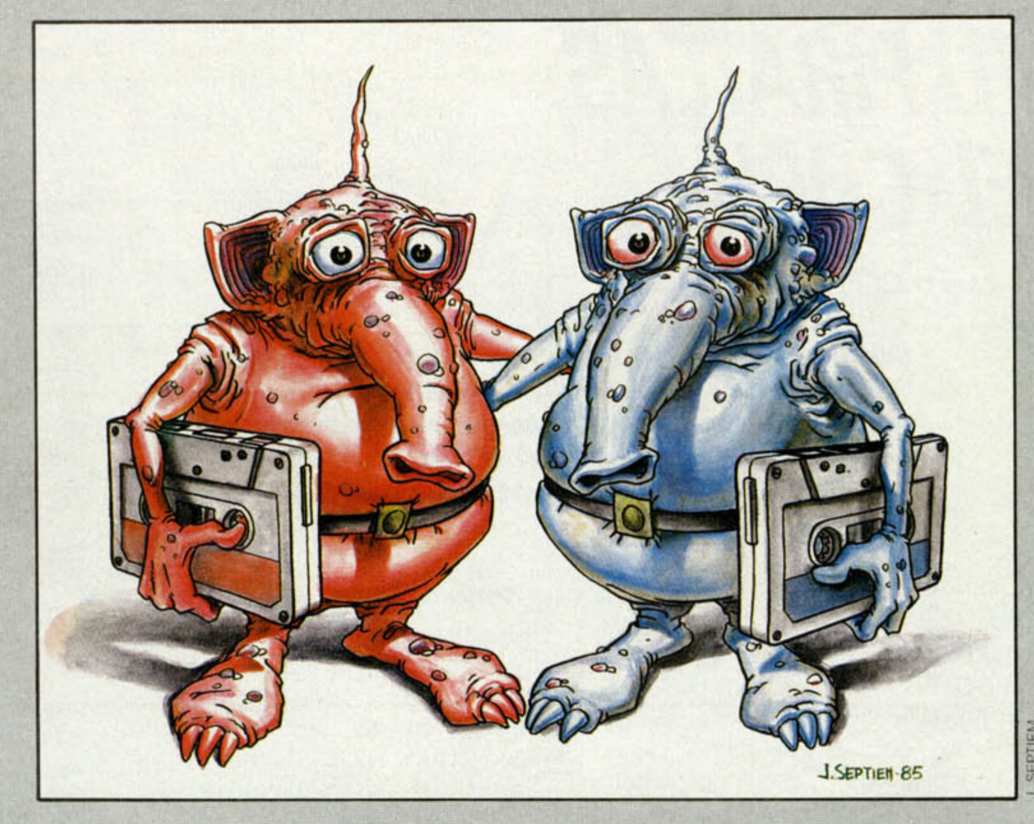
Supongamos que nuestro programa se llama «prueba». Para salvarlo en el drive teclearíamos:

(1) SAVE * "M"; 1; "PRUEBA"

y en vez de teclear ENTER, llevaríamos el cursor antes de la P de PRUEBA; ahora, pulsamos CAPS SHIFT + SYMBOL SHIFT y un número, por ejemplo, el "4". El papel cambiará a color verde; si queremos cambiar también la tinta, repetiremos la misma operación manteniendo pulsada CAPS SHIFT al teclear el número. Si lo hacemos con el "7" pondremos tinta blanca. Una vez hecho esto, llevamos el cursor después de las comillas del nombre del programa y restituimos los atributos originales por el mismo método. Ya podemos pulsar ENTER y salvar el programa. Para comprobar que todo ha salido correctamente, hacemos un CAT del drive y debiéramos ver el nombre del programa «PRUEBA» con el color de papel y tinta elegidos.

Si ahora tecleamos la línea (1) y pulsamos ENTER, salvaremos otra vez el programa, pero con distintos atributos, de forma que el sistema operativo del microdrive los considerará como ficheros distintos a pesar de tener al mismo nombre.

Una vez más, dejamos los posibles usos de esta técnica a la imaginación del lector, aunque una posibilidad sería utilizarla para tener copias de las distintas fases por las que pasa un programa que estemos desarrollando, hasta su conclusión final.



J. SEPTIEM

SIMULANDO LA CARGA DE PROGRAMAS

Si queremos incluir en nuestros propios programas el efecto de cargar programas desde la cassette, existe un grupo de direcciones en la memoria ROM que lo hacen con ruido de carga, sin ruido de carga, y con distintos colores a los habituales.

La pantalla tendrá la misma apariencia que cuando hacemos un LOAD desde el cassette, y permanecerá asi durante un tiempo variable que depende de la cantidad almacenada en ciertos registros del microprocesador.

Invitamos al lector a probar con las direcciones comprendidas entre la 1296 y la 1325, mediante la instrucción RANDOMIZE USR.

Si se cansa de esperar a que la rutina termine, pulse CAPS SHIFT + BREAK.

EJECUTANDO SENTENCIAS REM

Supongamos que en medio de un bloque de líneas de un programa BASIC necesitamos insertar una que no deseamos que se ejecute siempre, sino sólo en el momento en que a nosotros nos interesa; la forma normal de realizar esto consistiría en insertar en esa línea, o en otra parte del programa que llamará al bloque que estamos considerando, una expresión condicional del tipo IF THEN o similar, decidiendo si la línea debe o no ser ejecutada; este método, aunque funciona impecablemente, en según qué casos, puede complicar la estructura lógica del programa y relentizar su ejecución.

Un método alternativo sería colocar la línea «dudosa» en una sentencia REM, con lo cual nos aseguramos, de entrada, que no va a ser ejecutada.

En el momento y lugar del programa que interese ejecutar un salto a esa línea, pokearíamos en dos variables del sistema, llamadas NEWPIC y NSPPC, localizadas en las posiciones de memoria 23618 y 23620 respectivamente, las cuales representan el número de línea al que hay que saltar y el comando dentro de la línea que va a ser ejecutado.

```
  10 POKE 23618,100: POKE 23619,
0: POKE 23620,2
  20 STOP
 100 REM : PRINT "Esta linea se
ejecuta a pesar de todo."
```

En este espacio también tienen cabida los trucos que nuestros lectores quieran proponer.
Para ello, no tienen más que enviarlos por correo a MICROHOBBY, C/Arzobispo Morcillo, 24, of. 3 y 4, Madrid-28029.

YO PROTEJO, TU PIRATEAS, EL COPIA...

Lorenzo CEBEIRA

Más de una vez he oído decir que los usuarios del Spectrum nos dividimos en dos categorías: los que pasan noches en blanco ideando nuevos trucos para proteger sus programas contra miradas indiscretas, y los que pierden el sueño intentando hacer saltar estas protecciones.

No estoy seguro de a qué categoría pertenecen los lectores de MICROHOBBY, sospecho que a las dos, pero no me cabe duda que una buena parte debe tener unas hermosas ojeras causadas por estas apasionantes actividades. Espero que este pequeño artículo les permita recuperar alguna de las horas de sueño perdidas.

Cuando hablamos de protección de programas podemos referirnos a dos cosas: la primera consiste en evitar que éstos se puedan duplicar. En este terreno los trucos son infinitos aunque, por desgracia para las casas de software, un buen par de magnetófonos bastan para desarticular el más protegido de los programas.

Siempre quedará el purista que considera que este método no tiene mérito, pero la verdad es que sigue siendo el más utilizado. La última astucia de los fabricantes consiste en entregar el programa con una lista de claves necesarias para su funcionamiento, la cual está impresa en colores de poco contraste para dificultar la fotocopia. Sin embargo poca gente se asusta a la hora de copiar a mano esta lista si el programa merece la pena. Hasta tal punto han llegado las cosas en este terreno, que comienza a extenderse la creencia de que un programa fácil de copiar no debe ser muy bueno... En mi opinión, estas técnicas de protección son algo inútil, no conozco ninguna que pueda resistirse a un usuario medianamente avanzado, y por lo demás, mientras no se prohiba la tenencia de dos magnetófonos, no va a haber quien pare el fenómeno de la piratería de programas.

Protección de programas

En este estado de cosas más vale pasar al otro terreno de este campo de batalla: la protección de los programas destinada a evitar que puedan ser listados o modificados por el usuario.

En este género de protección tiene también un significado económico. En la mayor parte de los países, y el nuestro no es una excepción, la legislación sobre propiedad industrial e intelectual no protege, o protege muy mal, los derechos del autor de un algoritmo, de una idea para un videojuego, etc... Basta ver la cantidad de versiones que existen del famoso comecocos, para convencerse del perjuicio económico que supone esta falta de las legislaciones. Lógicamente, la reacción de los fabricantes consiste, una vez más, en acudir a los trucos de protección para evitar que otras personas puedan listar los programas y aprovecharse de las ideas y técnicas de los mismos. A diferencia de lo que ocurre en la protección contra copia, aquí sí es posible llegar a resultados bastante buenos, como vamos a ver a continuación. La otra cara de la moneda es que estas protecciones suponen, muy a menudo, un fastidio cuando nos interesa modificar un juego para aumentar, por ejemplo, el número de vidas, pasarlo a microdive, o para cualquier otro propósito legítimo. Es, pues, interesante conocer la panoplia de trucos más comunmente empleados.

Macedonia de trucos

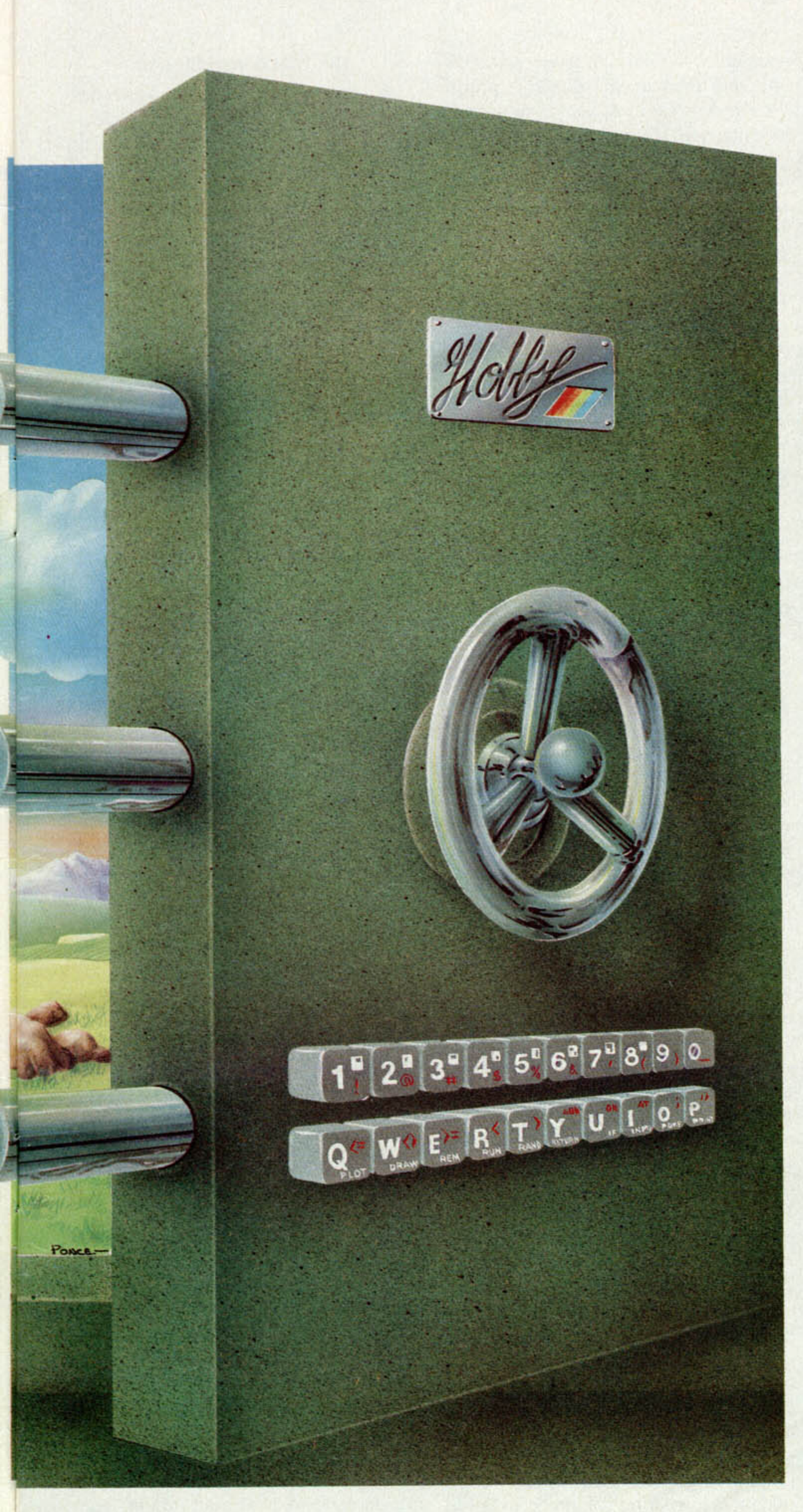
El primero y más conocido es el clásico POKE 23659,Ø que tiene por efecto bloquear nuestro ordenador si intentamos hacer un BREAK. La variable del sistema DF SZ cuya dirección es 23659 gestiona el número de líneas en la parte inferior de la pantalla. Si reducimos a cero este número, nuestro Spectrum no va a poder imprimir nada en esta zona de la pantalla y comoquiera que es precisamente aquí donde se van a imprimir los mensajes de error como el conocido «L BREAK into program» se producirá el clásico bloqueo que nos obligará a desenchufar nuestro ordenador o hacer uso del botón de reset en el caso de que lo hayamos instalado. El gran inconveniente de este truco es que no nos va a permitir utilizar la instrucción INPUT, por lo que si queremos introducir datos tendremos que hacerlo a través de la instrucción INKEY$.

Otra forma más elegante de obtener el mismo resultado es mediante POKE 23613,Ø que provocará un reset si intentamos parar el programa con la ventaja de no inutiizar la parte baja de la pantalla. En esta dirección 23613, se encuentra la variable ERR SP que indica al Spectrum dónde tiene que saltar en el caso de error y para nuestro ordenador, el BREAK o cualquier interrupción del programa se trata como un error. Modificando la dirección de salto forzamos la ejecución de la rutina de inicialización produciéndose el borrado total de la memoria.

Otra variante del mismo tema, más elegante todavía, es desactivar el BREAK recurriendo a la misma variable. En este caso, la instrucción a emplear es POKE 23613,82, con lo que en caso de BREAK forzamos un salto a la misma línea donde estaba ejecutándose el programa. El inconveniente de este truco es que no es permanente. Los bucles FOR-NEXT, los GOTO y los GOSUB-RETURN lo desactivan, por lo que es necesario repetir el POKE de marras, antes y después de cada instrucción de este tipo.

Otras innovaciones

Todos estos trucos tienen, sin embargo, un grave problema. Basta con cargar el programa en memoria, pero evitando su autoejecución, para tener libre acceso al listado, que es lo que se trata de evitar. Hace ya mucho tiempo que se des-



cubrió que basta con cargar un programa mediante la instrucción MERGE "" en vez de LOAD "", para evitar que éste empieze a funcionar, por mucho que lo hayamos grabado con SAVE " ... " LINE XXXXX. Pero como es lógico suponer, la alegría de los «piratas» duró poco tiempo. Rápidamente surgieron nuevas técnicas para evitar el MERGE. Las dos más conocidas consisten en grabar el programa como «Bytes» y provocar un bloqueo si se intenta utilizar el MERGE. Veámoslas con más detalle:

La primera de ellas requiere pararnos un poco a pensar en cómo funciona el intérprete BASIC de nuestro Spectrum. Cuando damos la orden de RUN el intérprete busca en las variables del sistema la dirección de comienzo del programa, toma nota del número de línea, la ejecuta y pasa a la siguiente. Cada vez que se termina de ejecutar una instrucción, las variables del sistema se actualizan para que en todo momento se sepa qué parte del programa está procesándose. Basándose en este principio, se pensó en grabar en el cassette el bloque de memoria que contiene no sólo el programa, sino también las variables del sistema, la zona de memoria donde se encuentran las direcciones de retorno para los GOSUB-RETURN y demás información necesaria para el intérprete basic. Si lanzamos la grabación durante la ejecución del programa y volvemos a cargarla a continuación, el intérprete continuará con la ejecución en la instrucción siguiente a la orden de SAVE. Veamos un ejemplo práctico con el siguiente programa:

Muchos han sido los métodos creados para proteger los programas, pero no menos han sido los «expertos» que han conseguido burlarlos.

```
9998 LET A=PEEK 23653 + 256 *
    PEEK 23654 : SAVE "programa"
    CODE 23552, A — 235ØØ
9999 GOTO "principio del programa"
```

Si colocamos estas líneas al final de nuestro programa y hacemos GOTO 9998 grabaremos en el cassette un bloque de «bytes» con toda la información

necesaria para que, una vez cargado, se ejecute automáticamente la línea 9999, que es la que lanzará nuestro programa. Y a ver quién es capaz de hacer un MERGE con un bloque de «bytes»!! Evidentemente es posible deshacer este truco, pero ya es harina de otro costal. Habría que cargar el bloque en la parte alta de memoria, protegido por RAMTOP, mediante un monitor de código máquina modificar «a mano» el listado del programa eliminando los trucos de protección que nos impidieran parar el programa así como las variables del sistema manipuladas, etc... En conjunto, un trabajo bastante pesado pero que, a veces, es necesario para pasar nuestros juegos favoritos al microdrive. Para los aficionados al código máquina, es más fácil copiar el bloque de memoria tal cual en el microdrive y cargarlo en una dirección alta de memoria donde no interfiera con las variables del sistema y, posteriormente, emplear la instrucción de código máquina LDIR para restituír el bloque a su lugar original. Si nuestros lectores estan interesados en el tema, les rogamos que nos lo hagan saber. Especialmente nos sería de gran utilidad saber cuántos disponen del microdrive para seguir profundizando en este terreno.

Antimerge

El segundo truco antiMERGE se basa en unos principios totalmente diferentes. Dado que esta instrucción está pensada originalmente para fusionar programas, es imprescindible que vaya tomando nota de los números de línea. El truco consiste en darle un número «imposible» con lo que la rutina se vuelve loca y se produce el consiguiente bloqueo. Naturalmente, hay que tener la precaución de restablecer las cosas a su estado normal antes de ejecutar el programa. Veamos un ejemplo:

```
1 POKE 238Ø2,Ø : POKE 23613,Ø
2 GOTO 4
3 POKE 238Ø1,255 : SAVE "programa"
LINE 1 : STOP
4 REM principio del programa
```

A la hora de proteger un programa contra MERGE incluiremos estas líneas al principio del mismo y lo grabaremos haciendo GOTO 3. El POKE de esta línea tiene por objeto transformar el número de la línea 2, cambiándolo a 65282, lo que para el Spectrum es imposible, y provocará el bloqueo si intentamos hacer MERGE. Por el contraio, si utilizamos el LOAD normal, la ejecución comenzará en la línea 1 donde el primer POKE devolverá a la línea 2 su número normal, y el segundo es uno de los de protección que ya hemos visto. Luego la ejecución se transferirá a la línea 4 donde comienza el resto del programa. Al utilizar esta rutinita hay que tener cuidado en no modificar la línea 1 ya que alteraríamos la dirección de la segunda línea con lo que el POKE debería de tener otra dirección. Por el contrario, si utilizamos microdive hay que tener en cuenta que la zona de programa no empieza en el sitio habitual y habrá que modificar las direcciones en consecuencia. La misma advertencia es de tener en cuenta para los siguiente trucos.

Inofensivo y divertido

Y siguiendo con nuestro panorama de trucos, vamos ahora con uno bastante inofensivo pero que siempre resulta divertido. El archifamoso de la línea Ø. Muchos programadores que conocen este truco, lo emplean para crear una línea que no se puede editar y donde colocan su nombre para evitar que otra persona se apropie del programa. No nos parece mal, pero rogamos que no lo pongan en los programas que nos envían ya que luego llueven cartas de lectores justamente indignados quejándose de que no pueden introducir esta línea en su ordenador. Para ello, basta con introducir la línea con un número normal y, posteriormente, modificarlo cambiando los dos primeros bytes de la misma mediante POKEs. Toda la dificultad consiste en conocer la dirección donde tenemos que POKEAr, si se me permite el barbarismo. Supongamos, por ejemplo, que en la primera línea de nuestro programa hemos puesto nuestro nombre. Basta en este caso, con hacer POKE 23756,Ø para transformarla en línea Ø que no se puede modificar mientras no hagamos otro POKE para deshacer el efecto. Si el número de esta línea fuese mayor que 255, entonces necesitamos añadir POKE 23755,Ø ya que estos números se codifican en dos bytes. Este truco se suele emplear en combinación con los caracteres de control de color para dejar el fondo del mismo color que la tinta con lo que el listado se vuelve invisible hasta que no editemos la línea Ø y eliminemos los carácteres de control. Recordemos que en el número 2 de MICROHOBBY se explica cómo variar los colores de las líneas del programa.

Un truco canallesco

Y vamos a terminar con un truco verdaderamente canallesco. Aquellos lectores que hayan tenido la paciencia de leer a fondo el manual del Spectrum (y hago aquí una pausa para afirmar que es uno de los mejores del mercado, otro día hablaremos de la traducción...) recordarán que los números se almacenan en memoria de dos formas. Primero viene la representación en ASCII del número, a continuación el código 14 que indica número y, por último, cinco bytes que lo representan en coma flotante. Lo interesante de este complicado asunto es que, para obtener el listado del programa, nuestro Spectrum emplea la representación el ASCII, mientras que para los cálculos emplea los cinco bytes mencionados. Esto significa que es posible cambiar los bytes que representan el número en ASCII dándoles unos valores irreales mientras que el programa va a seguir operando con los números reales, lo que nos permite confundir a todo aquel que intente reproducir nuestro programa. Y más interesante todavía, cuando editamos una línea «protegida» y la volvemos a introducir en memoria la representación en coma flotante cambia para ajustarse a la representación en ASCII lo que significa que el listado queda totalmente corrompido e inutilizable. Vamos con el ejemplo de turno:

Introduzcamos el siguiente y elemental programa:

```
1 PRINT 2 + 2
```

Y a continuación vamos a cambiar los números mediante POKE 2376Ø,49 y POKE 23768,49. Si ahora hacemos LIST obtendremos el siguiente programa:

1 PRINT 1 + 1, ya que 49 es el código ASCII del número 1.

Pero si hacemos RUN observaremos con gran sorpresa que el resultado sigue siendo 4, y no 2 como sería de esperar. Y por último, editemos la línea mediante CAPS SHIFT y EDIT y sin modificarla volvamos a introducirla con ENTER. Si ahora pulsamos nuevamente RUN, veremos que el resultado es 2, lo que significa que ha perdido definitivamente la representación en coma flotante de los números originales al intentar editar la línea. Combinando este truco con el de los caracteres de control de color para volver invisible nuestro programa, habremos conseguido producir un buen dolor de cabeza a todo aquel que intente meterse dentro del mismo.

¡Qué ustedes se lo pasen bien!

QUINIELAS

Spectrum 48 K

Amadeo CAMPOS

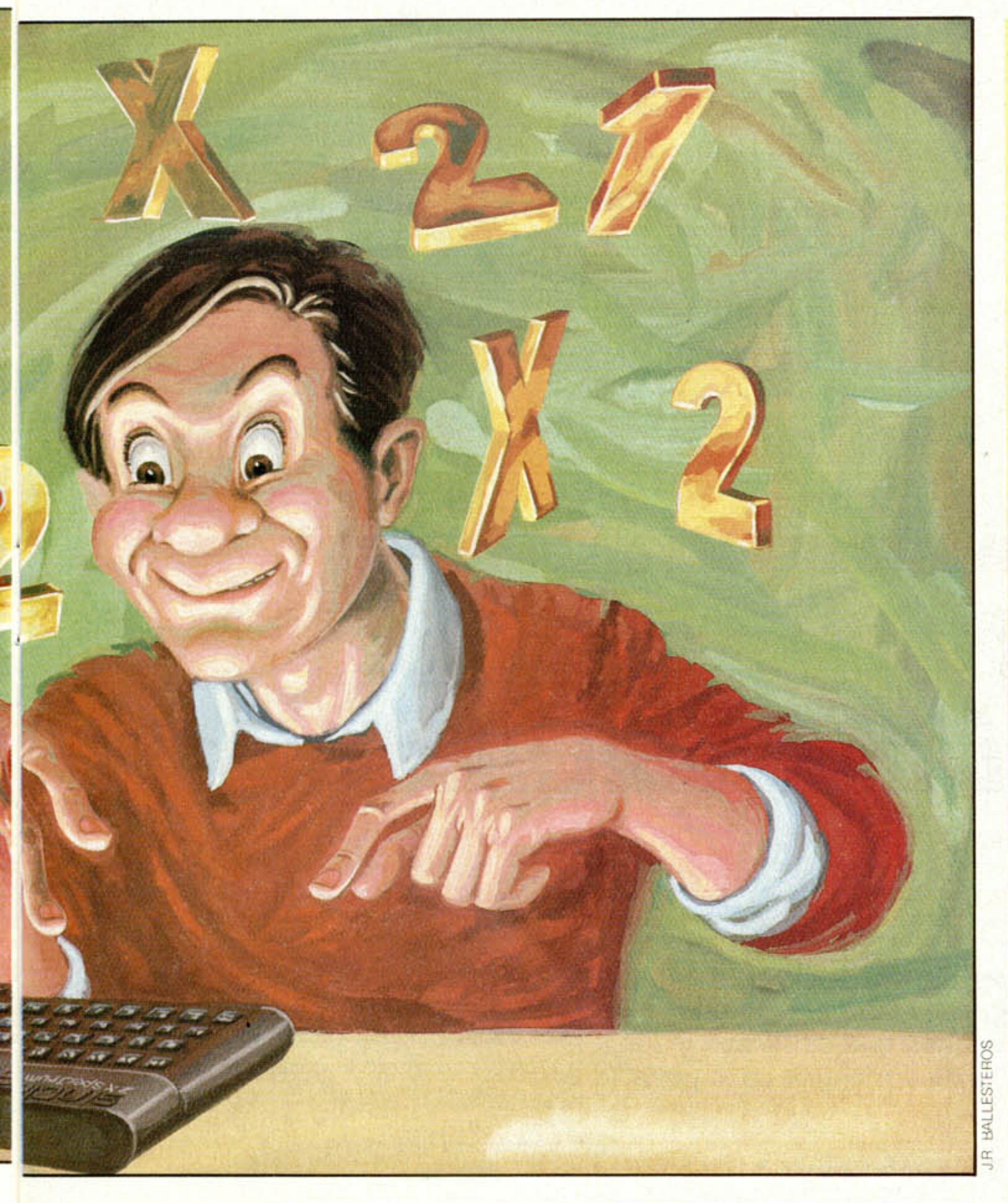
Si el fútbol es la «droga» nacional, como dicen, las quinielas son su antídoto. Así pues, dispongámonos a efectuar todo tipo de combinaciones y apuestas con este programa.

A pesar de que poco se puede decir de él que no quede lo suficientemente explicado en las múltiples pantallas que forman el programa, cabe hacer hincapié en el acierto de esta cinta que va a facilitar nuestras apuestas y nos va a ofrecer una serie de alternativas a la hora de combinar las diferentes opciones.

Pruébelo y decida por sí mismo.




```
   0>REM *****QUINIELAS POR***** AMADEO CAMPOS***REM Para hacer esta linea 0 la hice primero como linea 1 y luego POKE 23755,0 y POKE 23756,0.
   1 PAUSE 1: IF INKEY$="" THEN BORDER 1: BORDER 5: BORDER 1: BORDER 5: BORDER 1: BORDER 5: BORDER 7: GO TO 1
  10 RETURN : REM Efecto del borde en la presentacion de datos.
  20 BORDER 7: PAPER 7: INK 1: CLS
  30 BORDER 7: INK 1: PAPER 7: CLS
  35 PRINT AT 10,2;"QUIERES INSTRUCCIONES (S/N)?": GO SUB 1
  40 IF INKEY$="s" OR INKEY$="S" THEN GO TO 2230
  50 PAUSE 10: GO TO 1410
  60 BORDER 7: PAPER 7: INK 1: CLS : PRINT AT 1,12;"OPCIONES:"
  70 PRINT AT 3,2;"E...EMPEZAR A METER DATOS";AT 4,2;"2...SEGUIR METIENDO DATOS"
  80 PRINT AT 5,2;"3...FASE DE PRONOSTICOS"
  90 PRINT AT 6,2;"4...VER CLASIFICACION";AT 7,2;"5...ACTUALIZAR CLASIFICACION"
 100 PRINT AT 8,2;"6...VER PARTIDOS JUGADOS";AT 9,2;"P...STOP"
 110 PRINT AT 10,2;"L...'LOAD' DATOS DE CASSETTE";AT 11,2;"S...'SAVE' DATOS EN CASSETTE"
 120 PRINT AT 13,0; INK 1; PAPER 5; FLASH 1; BRIGHT 1;"████████████████████████████████";AT 14,8; INK 7; PAPER 2; FLASH 0;"ELIGE UNA OPCION";AT 15,0; INK 1; PAPER 5; FLASH 1;"████████████████████████████████"
 130 GO SUB 1
 140 LET m$=INKEY$
 150 IF m$="2" AND l(1)<>0 THEN GO TO 250
 160 IF m$="3" THEN GO TO 860
 170 IF m$="4" THEN GO TO 2110
 180 IF m$="5" THEN GO TO 1650
 190 IF m$="6" THEN GO TO 1920
 200 IF m$="P" THEN GO TO 2220
 210 IF m$="L" THEN CLS : PRINT AT 14,6;"PREPARADO PARA CARGAR": GO TO 1540
 220 IF m$="S" THEN GO TO 1260
 230 IF m$<>"E" THEN GO TO 60
 240 GO SUB 1390
 250 CLS
 260 DIM a(10): DIM b(10): DIM c(10)
 270 FOR j=1 TO 10
 280 LET a(j)=U(j): LET b(j)=X(j): LET c(j)=D(j)
 290 NEXT j
 300 LET e=25
 310 REM Ciclo de entrada de datos.
 320 PRINT INK 1;AT 20,2;"NUMERO DE DATOS:": INPUT datos
 330 CLS
 340 DIM U(10): DIM X(10): DIM D(10)
 350 FOR i=1+z TO z+datos
 360 CLS : PRINT INK 1;AT 18,5;"EQUIPOS:";AT 20,0;"CASA";AT 20,12;"VISITANTE": INPUT x,y
 370 IF x>equipos OR y>equipos THEN : PAUSE 10: PRINT AT 10,0;"ERROR EN EL NUMERO DEL EQUIPO!!!": GO SUB 1: GO TO 360
 380 LET l(i)=x: LET v(i)=y: LET f(i)=x: LET g(i)=y
 390 CLS
 400 PRINT INK 1;AT 20,0;"RESULTADO OBTENIDO (1-x-2):": INPUT LINE r$
 410 LET s$(i)=r$
 420 IF r$="" THEN GO TO 400
 430 IF r$(1)="p" THEN GO TO 0360: REM La letra 'p' es para corregir un dato.
 440 LET dp=-100*(x-y)/(e-1): REM  dp=diferencia de potencial.
 450 LET j=5+INT (dp/20)
 460 IF j>10 THEN LET j=10
 470 IF r$(1)="1" THEN LET U(j)=U(j)+1: GO TO 0510
 480 IF r$(1)="x" THEN LET X(j)=X(j)+1: GO TO 0510
 490 IF r$(1)="2" THEN.LET D(j)=D(j)+1: GO TO 0510
 500 BEEP .1,1: BEEP .1,1: BEEP .1,1: PRINT "ERROR EN ENTRADA DEL RESULTADO": PAUSE 200: CLS : GO TO 0400
 510 NEXT i: LET z=z+datos
 520 LET k=0
 530 LET t=0
 540 FOR j=1 TO 10
 550 LET t=total+U(j)+X(j)+D(j)
 560 LET total=t
 570 IF U(j)>k THEN LET k=U(j)
 580 IF X(j)>k THEN LET k=X(j)
 590 IF D(j)>k THEN LET k=D(j)
 600 NEXT j
 610 IF t<>z THEN PRINT "ERROR EN LA ENTRADA DE DATOS": GO TO 0350
 620 FOR j=1 TO 10: LET U(j)=U(j)+a(j): LET X(j)=X(j)+b(j): LET D(j)=D(j)+c(j): NEXT j
 630 GO TO 1560
 640 IF l(1)=0 THEN CLS : PRINT AT 10,9;"NO HAY DATOS": PAUSE 100: GO TO 60
 650 FOR j=1 TO 10: IF U(j)<>0 OR X(j)<>0 OR D(j)<>0 THEN LET B=10*j: GO TO 0670
 660 NEXT j
 670 LET l=INT (B/20): LET t=0: LET a=0: LET j=(B/10)-1
 680 LET P1=0: LET B1=0: LET Px=0: LET Bx=0: LET P2=0: LET B2=0
 690 LET B=(l+1)*20
 700 FOR d=-90+B TO 90-B STEP 20
 710 LET t=t+1: LET j=j+1: LET a=a+d*d
 720 LET S=a(j)+b(j)+c(j)+U(j)+X(j)+D(j)
 730 IF U(j)=0 AND X(j)=0 AND D(j)=0 THEN GO TO 800
 740 LET P1=P1+100*d*(U(j)+a(j))/S
 750 LET B1=B1+100*(U(j)+a(j))/S
 760 LET Px=Px+100*d*(X(j)+b(j))/S
 770 LET Bx=Bx+100*(X(j)+b(j))/S
 780 LET P2=P2+100*d*(D(j)+c(j))/S
 790 LET B2=B2+100*(D(j)+c(j))/S
 800 NEXT d
 810 LET A1=P1/a: LET Ax=Px/a: LET A2=P2/a
 820 LET B1=B1/t: LET Bx=Bx/t: LET B2=B2/t
 830 IF A1=0 OR Ax=0 OR A2=0 OR B1=0 OR Bx=0 OR B2=0 THEN PRINT AT 8,0;"DATOS INSUFICIENTES NECESITO MAS";AT 11,0;" PULSA UNA TECLA PARA CONTINUAR ": GO SUB 1: GO TO 60
 840 LET pro=1: IF m$="3" THEN GO TO 860
 850 GO TO 60
 860 IF l(1)=0 THEN CLS : PRINT AT 10,9;"NO HAY DATOS": PAUSE 100: GO TO 60
 870 IF pro=0 THEN CLS : PRINT AT 0,7;"ESPERA UN MOMENTO": GO TO 640
 880 CLS : PRINT AT 0,7;"FASE DE PRONOSTICOS"
 890 REM Probabilidades.
 900 REM Fase de pronosticos.
 910 PRINT AT 20,1;"NUMERO DE PRONOSTICOS ?": INPUT p
 920 CLS : PRINT " CONVIENE COPIAR LOS RESULTADOS ";"     QUE VAYAN APARECIENDO."
 930 FOR i=1 TO p
 940 PRINT
 950 INPUT INK 1;"EQUIPOS: ";x,y
 960 IF x>equipos OR y>equipos THEN PRINT AT 10,0;"ERROR EN EL NUMERO DEL EQUIPO!!!": GO SUB 1: CLS : GO TO 0950
 970 LET dp=-100*(x-y)/(e-1)
 980 IF dp<-100+B THEN LET dp=-100+B
 990 IF dp>100-B THEN LET dp=100-B
1000 LET P1=A1*dp+B1
1010 IF P1>100 THEN LET P1=100
1020 IF P1<0 THEN LET P1=0
1030 LET Px=Ax*dp+Bx
1040 IF Px>100 THEN LET Px=100
1050 IF Px<0 THEN LET Px=0
1060 LET P2=A2*dp+B2
1070 IF P2>100 THEN LET P2=100
1080 IF P2<0 THEN LET P2=0
1090 LET T=P1+Px+P2
1100 LET P1=100*P1/t: LET Px=100*Px/t: LET P2=100*P2/t
1110 REM Las tres lineas siguientes es para eliminar algunos decimales.
1120 LET p$=STR$ P1: LET w$=STR$ Px: LET y$=STR$ P2
1130 IF LEN p$<5 THEN LET p$=p$+"    "
1140 IF LEN w$<5 THEN LET w$=w$+"    "
1150 IF LEN y$<5 THEN LET y$=y$+"    "
```



J.R. BALLESTEROS

```
1160 PRINT " P1=";p$( TO 5);" PX
=";w$( TO 5);" P2=";y$( TO 5)
 1170 NEXT i
 1180 PRINT
 1190 PRINT
 1200 PRINT " DESEA OBTENER MAS P
 RONOSTICOS                (s/n)?
 ": GO SUB 1
 1210 IF INKEY$="s" OR INKEY$="S"
  THEN GO TO 0910
 1220 CLS : PRINT AT 10,2;"METER
 DATOS DE OTRA CATEGORIA
      (S/N)"
 1230 GO SUB 1
 1240 IF INKEY$="s" OR INKEY$="S"
  THEN RUN 1410
 1250 CLS : PRINT AT 11,3;"BUENA
 SUERTE EN LA JORNADA"
 1260 PRINT AT 13,1;"PREPARA UNA
 CINTA PARA GRABAR"
 1270 PRINT AT 15,1;"PULSA UNA TE
 CLA CUANDO QUIERAS"
 1280 GO SUB 1
 1290 LET h$=STR$ equipos
 1300 SAVE "clasif. "+h$(1) DATA
 f$(): SAVE "local "+h$(1) DATA l
 (): SAVE "visit. "+h$(1) DATA v(
 ): SAVE "result. "+h$(1) DATA s$
()
1310 CLS : PRINT AT 8,12; FLASH
1;"GRABADO!": PAUSE 100: PRINT A
T 10,3; FLASH 0;"QUIERES VERIFIC
ARLO? (S/N)": PAUSE 0
1320 IF INKEY$="s" OR INKEY$="S"
 THEN GO TO 1340
1330 GO TO 1380
1340 PAPER 1: CLS
1350 PRINT AT 10,10; FLASH 1;"VE
RIFICANDO": PAPER 1: INK 1: PRIN
T AT 0,0
1360 VERIFY "clasif. "+h$(1) DAT
A f$(): VERIFY "local "+h$(1) DA
TA l(): VERIFY "visit. "+h$(1) D
ATA v(): VERIFY "result. "+h$(1)
 DATA s$()
```

```
1370 PRINT AT 10,10; FLASH 1;"
CORRECTO!!": PAUSE 100
1380 GO TO 60
1390 LET z=0: LET t=0: LET total
=0: DIM U(10): DIM X(10): DIM D(
VALLADOLID","ZARAGOZA","ELCHE","
MURCIA"
1490 CLS : PRINT AT 10,3;"QUIERE
S CARGAR DATOS (S/N)?": GO SUB 1
1500 IF INKEY$="s" OR INKEY$="S"
 THEN GO TO 1520
1510 GO TO 60
1520 CLS : PRINT AT 0,2;"PREPARA
DO PARA CARGAR DATOS."
1530 PRINT AT 3,4;"PULSA PLAY EN
 EL CASSETTE"
1540 LET h$=STR$ equipos: LOAD "
clasif. "+h$(1) DATA f$(): LOAD
"local "+h$(1) DATA l(): LOAD "v
isit. "+h$(1) DATA v(): LOAD "re
sult. "+h$(1) DATA s$().
1550 CLS : PRINT AT 10,11;"GRABA
DO": PAUSE 100
1560 FOR i=1 TO 800: IF l(i)=0 A
ND v(i)=0 THEN LET z=i-1: LET to
tal=z: GO TO 1790
1570 NEXT i

10)
1400 RETURN
1410 INK 1: LET pro=0: LET e=25:
 DIM t(800): DIM f(800): DIM g(8
00): DIM l$(20,10): DIM s$(800):
 DIM l(800): DIM v(800)
1420 DIM f$(20,10)
1430 CLS : PRINT AT 10,1;"EQUIPO
S DE PRIMERA O SEGUNDA ?": PRINT
 AT 11,13;"1 o 2": GO SUB 1
1440 IF INKEY$="2" THEN LET equi
pos=20: GO SUB 2190: GO TO 1490
1450 IF INKEY$="1" THEN LET equi
pos=18: GO TO 1470
1460 GO TO 1430
1470 RESTORE 1480: FOR i=1 TO eq
uipos: READ l$(i): NEXT i
1480 DATA "BARCELONA","VALENCIA"
```

```
,"R.MADRID","GIJON","AT.MADRID",
"SEVILLA","R.SOCIEDAD","OSASUNA"
,"AT.BILBAO","HERCULES","BETIS",
"SANTANDER","MALAGA","ESPANOL","
1580 GO TO 1790
1590 PRINT AT 10,7;"DATOS ACTUAL
IZADOS.": PRINT #1;"PULSA UNA TE
CLA PARA VER MENU.": GO SUB 1: G
O TO 60
1600 IF l(1)=0 THEN CLS : PRINT
AT 10,9;"NO HAY DATOS": PAUSE 10
0: GO TO 60
1610 CLS : PRINT AT 0,6;"PARTIDO
S JUGADOS:": PAUSE 50: FOR i=1 T
O z: PRINT TAB 0;l(i);TAB 3;l$(l
(i));TAB 14;v(i);TAB 17;v$(v(i))
;TAB 28;s$(i)
1620 NEXT i
1630 PRINT #1;"PULSA UNA TECLA P
ARA VER MENU.": GO SUB 1: GO TO
60
1640 CLS : IF l(1)=0 THEN CLS :
PRINT AT 10,9;"NO HAY DATOS": PA
USE 100: GO TO 60
1650 CLS : DIM t(800): FOR n=1 T
O equipos
1660 PRINT AT n-1,1;l$(n);" LUGA
R EN LA TABLA: ": INPUT k
1670 IF k>equipos THEN GO TO 170
0
1680 FOR i=1 TO n: IF t(i)=k THE
N GO TO 1700
1690 NEXT i: GO TO 1710
1700 PRINT #1;"ERROR EN EL NUMER
O DEL EQUIPO!!!": PRINT AT n-1,3
0; FLASH 1;k: GO SUB 1: PRINT #1
;"
  ": PRINT AT n-1,30;"  ": GO TO
 1660
1710 LET t(n)=k: PRINT AT n-1,30
; INK 0;k
1720 LET f$(k)=l$(n)
1730 NEXT n
1740 PRINT #1;"ESTA LA CLASIFICA
CION BIEN (S/N)": GO SUB 1
1750 IF INKEY$="s" OR INKEY$="S"
 THEN GO TO 1790
1760 IF INKEY$="n" OR INKEY$="N"
 THEN GO TO 1650
1770 GO TO 1750
1780 IF l(1)=0 THEN GO TO 1910
1790 CLS : PRINT AT 10,7;"ACTUAL
IZANDO DATOS"
1800 IF l(1)=0 THEN LET z=1
1810 FOR i=1 TO equipos
1820 FOR n=1 TO z
1830 PRINT AT 2,0;"TOTAL E.: ";eq
uipos;AT 2,12;"TOTAL D.: ";equipo
s;"*";z;"=";z*equipos
1840 IF l(n)=i THEN LET f(n)=t(i
)
1850 IF t(i)=0 THEN LET f(n)=l(n
)
1860 IF v(n)=i THEN LET g(n)=t(i
)
1870 IF t(i)=0 THEN LET g(n)=v(n
)
1880 LET i$=STR$ i: LET n$=STR$
n: PRINT AT 0,0;"EQUIPOS:  ";AT 0
,11-LEN i$;i;AT 0,12;"DATOS:  ";A
T 0,22-LEN n$;n: IF n=z THEN PRI
NT AT 0,18;"
1890 NEXT n
1900 NEXT i
1910 GO SUB 1970: GO TO 60
1920 IF f(1)=0 THEN CLS : PRINT
AT 10,9;"NO HAY DATOS": PAUSE 15
0: GO TO 60
1930 CLS : PRINT AT 0,6;"PARTIDO
S JUGADOS:": PAUSE 50: FOR n=1 T
O z
1940 PRINT TAB 0;l(n);TAB 3;f$(l
(n));TAB 15;v(n);TAB 18;f$(v(n))
;TAB 30;s$(n)
1950 NEXT n
1960 PRINT #1;"PULSA UNA TECLA P
ARA VER MENU.": GO SUB 1: GO TO
60
1970 DIM a(10): DIM b(10): DIM c
(10): DIM U(10): DIM X(10): DIM
D(10)
1980 IF l(1)=0 THEN GO TO 2100
1990 FOR i=1 TO z
2000 LET r$=s$(i)
2010 IF f(i)=0 AND g(i)=0 THEN L
ET x=l(i): LET y=v(i): GO TO 203
0
2020 LET x=f(i): LET y=g(i)
2030 LET dp=-100*(x-y)/(e-1)
2040 LET j=5+INT (dp/20)
2050 IF j>10 THEN LET j=10
2060 IF r$="1" THEN LET U(j)=U(j
)+1: GO TO 2090
2070 IF r$="x" THEN LET X(j)=X(j
)+1: GO TO 2090
2080 IF r$="2" THEN LET D(j)=D(j
)+1: GO TO 2090
2090 NEXT i
2100 RETURN
2110 IF f$(1)<>"          " THEN
 GO TO 2130
2120 CLS : PRINT AT 10,9;"NO TEN
GO DATOS";AT 13,4;"EMPIEZA POR L
A OPCION 5": GO SUB 1: GO TO 60
2130 CLS : PRINT AT 0,3;"CLASIFI
CACION ACTUAL.": FOR n=1 TO equi
pos: IF f$(1)<>"          " THEN
 GO TO 2150
2140 PRINT TAB 1;n;TAB 4;l$(n):
GO TO 2160
2150 PRINT TAB 1;n;TAB 4;f$(n)
2160 NEXT n
2170 PRINT #1;"PULSA UNA TECLA P
ARA VER MENU."
2180 GO SUB 1: GO TO 60
2190 RESTORE 2200: FOR i=1 TO 20
: READ l$(i): NEXT i
2200 DATA "LAS PALMAS","CADIZ","
CASTILLA","CELTA","LOGRONES","SA
BADELL","MALLORCA","CASTELLON","
```

```
HUELVA","BARCEL.AT.","BILBAO AT.
","CARTAGENA","TENERIFE","D. COR
UÑA","AT.MADRIL.","GRANADA","SAL
AMANCA","LORCA","OVIEDO","CALVO
SOT."
2210 RETURN
2220 STOP
2230 PAUSE 10: PRINT AT 0,8; PAP
ER 6; INK 0; BRIGHT 1;"**INSTRUC
CIONES**"
2240 PRINT : PRINT " Primero tie
nes que numerar los equipos segu
n el orden en la ta-bla de clasi
ficacion OPCION 5"
2250 PRINT : PRINT " El programa
 empieza presentando";"un serie
de opciones elige una."
2260 PRINT : PRINT " Si eliges l
a opcion E o la ■ te";"hara algu
nas preguntas como son:"
2270 PRINT : PRINT " NUMERO DE D
ATOS:...Contesta el numero de pa
rtidos cuyo resulta-do conoces."
2280 PRINT : PRINT " EQUIPOS:...
Con la distincion de";"CASA y VI
SITANTE: Contesta por ";"el num
ero que le has asignado segun la
 ultima clasificacion."
2290 PRINT : PRINT " Para pasar
pagina pulsa tecla."
2300 GO SUB 1
2310 CLS : PRINT AT 0,8; PAPER 6
; INK 0; BRIGHT 1;"**INSTRUCCION
ES**"
2320 PRINT : PRINT " RESULTADO O
BTENIDO: (1-X-2)....Contesta el
que se haya dado en cada partido
,es decir, 1 x 2."
2330 PRINT : PRINT " La diferenc
ia entre la opcion E y la ■ es q
ue en la primera em-piezan todos
 los datos desde ce-ro y en la s
egunda se conservan datos metido
s anteriormente."
2340 PRINT : PRINT " Esta fase s
e repetira tantas veces como num
ero de datos hayas  contestado."
2350 PRINT : PRINT
2360 PRINT " Para pasar pagina p
ulsa tecla."
2370 GO SUB 1
2380 CLS : PRINT AT 0,8; PAPER 6
; INK 0; BRIGHT 1;"**INSTRUCCION
ES**"
2390 PRINT : PRINT " Seguidament
e se pasa a la parte";"de pronos
ticos, en la que pregun";"tara p
or el numero de ellos que ";"qui
eres conocer y los equipos   ";"
que quieras pornosticar."
2400 PRINT : PRINT AT 8,8;"**OBS
ERVACIONES**"
2410 PRINT : PRINT " Ten mucho c
uidado al meter los datos para n
o equivocarte."
2420 PRINT " Si te equivocas al
meter un da-to se puede corregir
 contestando 'p' en la pregunta
RESULTADO OBTENIDO: ."
2430 PRINT : PRINT " Cuando se a
cabe de ejecutar to-do el progra
ma, este se grabara (SAVE) solo
para conservacion de los datos."
2440 PRINT " Para pasar pagina p
ulsa tecla."
2450 GO SUB 1
2460 CLS : GO TO 1410
3000 SAVE "1-x-2" LINE 30
```

DIBUJAR EN TRES DIMENSIONES

Fernando ROMAY RODRIGUEZ

Spectrum 16 K

Si quiere dibujar en tres dimensiones, este es, sin lugar a dudas, su programa. Para conseguirlo, revise atentamente las instrucciones que se le dan y conseguirá transformar cualquier objeto en la pantalla de su Spectrum.

En primer lugar, hemos de dar al programa las coordenadas de los puntos que constituyen el objeto en el espacio y, más tarde, las conexiones de los puntos del mismo, teniendo en cuenta que cada punto lo podemos conectar con otros dos. Esto se hace con la opción uno del menú, la de crear datos.

Una vez introducidos los puntos, podremos variar la escala, la posición del objeto en la pantalla e, incluso, verlo desde cualquier ángulo al poderlo girar en torno a los tres ejes.

Supongamos que queremos dibujar un cubo, por ejemplo. Lo primero que hay que hacer es introducir la opción 5 para fijar los colores, borde, papel y tinta. Seguidamente, introduciremos la opción 1 con los datos del dibujo (nos preguntará el número de puntos y con cuáles de ellos debemos conectar cada uno de los ocho). Terminadas las conexiones, se nos preguntará si son correctas, y si todo está bien, el programa presentará el menú.

El otro paso a seguir es la introducción de la opción 4, para dibujar. En este momento nos aparecerá el mensaje «pulsa m para menú»:

Si introducimos m volveremos al menú.

Si introducimos a podremos cambiar alfa, girar en torno a x.

Si introducimos b podremos cambiar beta, girar en torno a y.

Si introducimos g podremos cambiar gamma, girar en torno a z.

Si introducimos e podremos cambiar la escala,

y así para las demás alternativas que tiene la opción 4.

Para dibujar después de haber realizado un cambio de alfa, beta, etc., no hace falta volver al menú, es suficiente con pulsar ENTER.

Al introducir para alfa = 9Ø y beta = 6Ø y gamma = 6Ø con escala = 5Ø, veremos un bonito *cubo en 3 dimensiones*.

Para grabar este dibujo, introducimos la opción 2 y sólo tendremos que dar el nombre de la figura, con lo que se graba en cinta. De esta forma podemos crear gran cantidad de dibujos.

Para cargarlo basta introducir la opción 3. Y dar el nombre.

```
   1 LET t=7
   5 DIM p(3,2): LET f=0: LET b=
f: LET g=b
   6 LET e=40
   7 LET cx=128: LET cy=88
  15 CLS
  20 PRINT "Menu:"
  22 PRINT "       1-Crear datos."
  24 PRINT "       2-Grabar datos.
"
  26 PRINT "       3-Leer datos."
  28 PRINT "       4-Dibujar."
  29 PRINT "       5-Papel,Borde..
.": PRINT : PRINT "      En opcio
n 4:": PRINT : PRINT "     a-Gir
o segun x.": PRINT "     b-Giro
segun y.": PRINT "     g-Giro se
gun z.": PRINT "     e-Escala.":
 PRINT "      x-Centro x": PRINT
"      y-Centro y"
  30 INPUT FLASH 1;"INTRODUCE OP
CION"; FLASH 0;,A
  32 IF A<1 OR A>5 THEN GO TO 30
  40 GO TO a*100
 100 INPUT "Numero de datos ?";n
: DIM a(n+1,5): LET a(1,1)=n
 102 FOR i=2 TO n+1: FOR j=1 TO
3: INPUT ("Punto:";i-1;",";"Coor
denada:";CHR$ (j+119);"  ");a(i,
j): NEXT j: NEXT i
 104 CLS : PRINT "Son correctos:
 s/n ?": INPUT a$: IF a$(1)="s"
OR a$(1)="S" THEN GO TO 106
 105 CLS : INPUT "Punto:",p: INP
UT "Coordenada:1,2,3 para x,y,z
",c: INPUT ("Punto:";p;",";"Coor
denada:";CHR$ (c+119);"  ");a(p+1
,c): GO TO 104
 107 CLS : PRINT "Ahora las cone
xiones:": FOR i=2 TO n+1
 108 INPUT ("El punto:";i-1;" co
n:?");a(i,4): INPUT "Y con:?";a(
i,5): IF a(i,4)<0 OR a(i,4)>n OR
 a(i,5)<0 OR a(i,5)>n THEN GO TO
 108
 111 NEXT i
 112 CLS : PRINT "Son correctas
```

```
las conexiones: s/n ?": INPUT a$
: IF a$(1)="s" OR a$(1)="S" THEN
 GO TO 116
 114 INPUT "Punto:";c
 115 INPUT ("El punto:";c;" con:
?");a(c+1,4): INPUT "Y con:?";a(
c+1,5): GO TO 112
 130 GO TO 15
 200 CLS : INPUT "Nombre del fic
hero:";a$
 202 PRINT "Atento a pulsar..."
 204 SAVE a$ DATA a():
 206 GO TO 15
 300 CLS : INPUT "PULSA PLAY,Nom
bre ? ";a$
 302 LOAD a$ DATA a():
 303 LET n=a(1,1)
 304 GO TO 15
 400 CLS : LET p(1,1)=(SIN g*COS
 f-COS g*SIN b*SIN f)*e
 401 LET p(1,2)=(COS g*SIN b*COS
 f+SIN g*SIN f)*e
 402 LET p(2,1)=(COS g*COS f+SIN
 g*SIN b*SIN f)*e
 403 LET p(2,2)=(COS g*SIN f-SIN
 g*SIN b*COS f)*e
 404 LET p(3,1)=(-SIN f*COS b)*e
 405 LET p(3,2)=COS f*COS b*e
 410 FOR i=2 TO n+1
 411 LET y=a(i,5)+1: LET x=a(i,4
)+1
 412 LET x1=a(i,1)*p(1,1)+a(i,2)
*p(2,1)+a(i,3)*p(3,1)+cx
 413 LET y1=a(i,1)*p(1,2)+a(i,2)
*p(2,2)+a(i,3)*p(3,2)+cy
 414 LET x2=a(x,1)*p(1,1)+a(x,2)
*p(2,1)+a(x,3)*p(3,1)+cx
 415 LET y2=a(x,1)*p(1,2)+a(x,2)
*p(2,2)+a(x,3)*p(3,2)+cy
 416 LET x3=a(y,1)*p(1,1)+a(y,2)
*p(2,1)+a(y,3)*p(3,1)+cx
 417 LET y3=a(y,1)*p(1,2)+a(y,2)
*p(2,2)+a(y,3)*p(3,2)+cy
 420 PLOT x1,y1: DRAW x2-x1,y2-y
1: PLOT x1,y1: DRAW x3-x1,y3-y1
 421 NEXT i
 422 PRINT INK 9;AT 21,0;"a=";f*
180/PI;"' b=";b*180/PI;" g=";g*18
0/PI;"
 423 INPUT "Menu: pulsa m",a$: I
F a$="m" THEN GO TO 15
 425 IF a$="a" THEN INPUT "alfa:
",f: LET f=f*PI/180
 426 IF a$="b" THEN INPUT "beta:
",b: LET b=b*PI/180
 427 IF a$="g" THEN INPUT "Gamma
:",g: LET g=g*PI/180
 428 IF a$="e" THEN INPUT "Escal
a:",e
 429 IF a$="x" THEN INPUT "Centr
o x:",cx
 430 IF a$="y" THEN INPUT "Centr
o y:",cy
 435 IF a$<>"" THEN GO TO 423
 438 GO TO 400
 500 CLS : INPUT "Border:",c: BO
RDER c: INPUT "Papel:",c: PAPER
c: INPUT "Tinta:",t: INK t: GO T
O 15
 600 SAVE "3d" LINE 1
```



Viajando a través del tiempo

KOKOTONI WILF

Dentro de los programas de Arcade, KOKOTONI WILF es uno de los más divertidos. La ambientación de sus pantallas, la dificultad del juego, y la simplicidad de movimientos, hacen de él un programa con un nivel de adicción muy alto.



48 K

Tipo de juego: Arcade

Inglés

Dentro de la amplia gama de juegos de Arcade que existe actualmente en el mercado, es necesario hacer una diferenciación entre aquellos en los que es necesario mantener un duelo de habilidad y reflejos contra el ordenador, y otros en los que además de habilidad, es necesario realizar un estudio exhaustivo de la situación antes de movernos, ya que las pantallas que hay que atravesar nos deparan imnumerables sorpresas. Kokotroni Wilf es de estos últimos.

El objetivo, en esta ocasión, es conseguir un determinado número de llaves en cada una de las diferentes partes del juego. Cada una de estas fases nos sitúa en una época concreta que va, desde la era primitiva, pasando por las civilizaciones antiguas, hasta llegar a las épocas más modernas. Nuestro personaje, Kokotoni Wilf, tiene que encontrar todas las piezas del amuleto del dragón que están ocultas en algún lugar de la historia, a lo largo del tiempo. Las piezas han sido escondidas por su maestro, el gran



mago «ULRICH».

Para conseguir llevar a cabo la misión, habrá que luchar contra todo tipo de enemigos, desde los dinosaurios de la época prehistórica hasta los robots de la era espacial.

Los amuletos que hay que buscar tienen forma de estrella. En cada época por la que pasemos tendremos que buscar, en primer lugar, un determinado número de éstas. Cuando lo hayamos logrado aparecerá entonces una de las estrellas intermitentes en alguno de los lugares de dicha época





por los que hemos pasado. Es necesario que la encontremos y la cojamos, ya que va a ser la llave que nos permita poder pasar a otra nueva época. Este proceso se irá repitiendo durante el juego en varias ocasiones, aumentando el nivel de dificultad a medida que avanzamos a una época diferente.

El juego está en la línea de Maic Miner y Jet Set Willy, en los que, recordemos, teníamos que ir recogiendo unas llaves en cada habitación para poder pasar a la siguiente. En esta ocasión, el programa tiene



sesenta pantallas de alta resolución.

Gráficamente se trata de un gran juego, los sprites son de gran calidad y llegan a asemejarse a los dibujos animados. El movimiento es bastante simple y muy fácil de controlar a través del teclado. Básicamente, existen tres únicos controles: para ir a la derecha, a la izquierda y uno que nos permite volar, (recordemos que Kokotoni es una especie de angelito).

Es un juego muy entretenido y con un nivel de dificultad alto que nos mantendrá delante de la pantalla largas horas tratando de conseguir nuestro objetivo. Una ventaja, sin embargo, que tiene el juego, es el hecho de que si cometemos un fallo no nos retorna al principio, sino que empezamos de nuevo en la



última posición en la que nos encontrábamos.

La ambientación de cada época es muy atractiva, dota al juego de un interéss especial y de un deseo incontenible del jugador de averiguar cómo será la siguiente época y qué peligros encontrará en ésta. Muy entretenido.

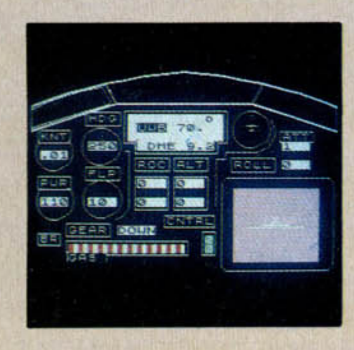
Pilota tu propio avión



Protec/ABC
48 K
Tipo de juego: Simulador de vuelo
PVP: 1.695



Este programa es un simulador de vuelo, con las características de un avión comercial. Por medio de los controles de los que disponemos en el juego, podemos despegar, maniobrar y aterrizar, según la fase del juego en la que nos encontremos.
La posición en la que nos hallamos se puede visualizar a través de un mapa, o bien cuando nos dispongamos a aterrizar, momento en el cual podremos ver la pista de aterrizaje.
El programa nos permite la posibilidad de utilizar un joystick; de este modo conseguimos un efecto mayor de realismo y, a la vez, se facilita bastante el manejo de un grupo grande de controles.
Nada más cargar el programa, aparece en la imagen del televisor un panel de instrumentos en el cual se encuentran todos los dispositivos necesarios para manejar el aparato.
Dicho panel consta de trece controles diferentes: indicador de velocidad del aire; visualizador de ruta; medidor de potencia; indicadores de alerones, frenos, tren de aterrizaje, de combustible y de control; medidor del grado de ascenso y descenso; altímetro, visualizador de proa; medidores de posición y giro y panel de navegación.
Al principio, se nos pregunta si queremos empezar directamente a volar o si queremos efectuar un despegue desde la pista. Si elegimos la opción de vuelo, podemos indicar la altura a la que queremos volar. El viento será decisivo a la hora de lograr mantener el rumbo, ya que puede desviarnos de la ruta prevista si no tenemos mucho cuidado. Es posible definir la posición de partida, para lo cual habrá que haber definido previamente una baliza, la posición radial, la dirección de la proa del avión y la distancia desde la baliza escogida. Se puede, de este modo, también hacer prácticas de aterrizaje si definimos los parámetros adecuados, los cuales nos son mostrados en las instrucciones del juego.
La mayor parte del programa está escrito en Basic, exceptuando alguna rutina en código máquina, a pesar de ello, no desmerece en absoluto, sobre todo si tenemos en cuenta que la compañía Proteck es una experta en este tipo de juegos.



Un bosque peligroso



Micromega/Ventamatic
48 K
Tipo de juego: Arcade
PVP: 1.700



Dentro de los denominados juegos de acción, Deathchase es un de los más espectaculares por la forma en la que ha sido construido. La idea en la que está basado es bastante simple, nosotros dirigimos una moto espacial y tenemos que lograr impedir que los enemigos surquen nuestro territorio; sin embargo, a pesar de ello, el juego resulta muy entretenido.
Hay nueve fases en cada una de las cuales el nivel de dificultad aumenta.
Nuestra misión es la de dirigir el vehículo de que disponemos en busca de los supuestos invasores.
Para conseguirlo será necesario tener mucho cuidado, ya que el planeta que vigilamos está lleno de árboles que dificultan nuestro recorrido y pueden provocar un choque.
Cada nivel tiene un número mayor de árboles, por lo que en los últimos, resulta francamente difícil conseguir nuestro objetivo.
Los enemigos a los que tenemos que enfrentarnos llevan unas motos e intentan escabullirse por el bosque para impedir que les demos caza. También hay helicópteros que recorren nuestros dominios y a los que tenemos que tratar de destruir.
La presentación del programa es buena, y los gráficos están bastante conseguidos. La sensación de velocidad se hace patente durante todo el programa, sobre todo si se tiene en cuenta que el juego ha sido realizado en tres dimensiones y, por cierto, bastante bien. Cuando vamos avanzando por el bosque, los árboles se van haciendo más grandes a medida que nos acercamos a ellos. Si chocamos contra uno de ellos, el efecto del golpe está muy logrado.
Nuestra misión dura día y noche, cuando llega esta última cambia el decorado de la pantalla y se oscurece el fondo.
Los efectos sonoros son muy buenos y la ambientación nos recuerda una escena de la Guerra de las Galaxias: concretamente a la persecución de la tercera parte de dicha serie.
Es un juego entretenido en el que la sensación de perspectiva y velocidad nos van a hacer vibrar aunque sólo sea mientras dure éste.

Localiza a tu enemigo

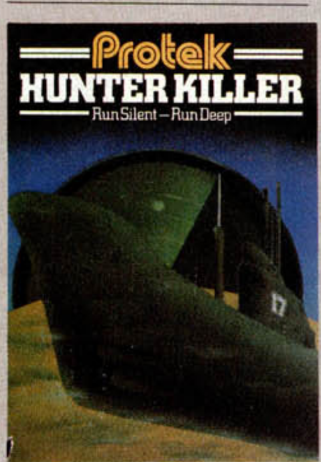
HUNTER KILLER

Protec/ABC

48 K

Tipo de juego: Estrategia

PVP.: 2.250



Hunter-Killer, es un juego simulador de combate, que tuvo un gran éxito en Inglaterra. La revista Sinclair User, en su sección de programas, le puntuaba con un nueve, lo cual, tras lo visto, nos parece justo.

El objetivo del juego es dirigir un submarino en busca del enemigo, tratando de que éste no logre darnos alcance y evitando ser hundido por las cargas de profundidad que en el transcurso del juego lanza su aviación, para lo cual disponemos de sofisticados mecanismos de defensa.

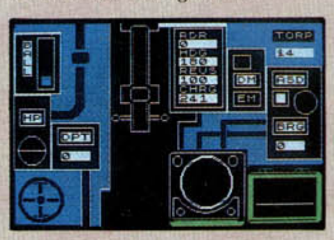
Una vez cargado el juego, lo primero que aparece en la pantalla es una especie de plano de situación, junto a un mensaje que indica la latitud y la longitud donde se encuentra el submarino. Seguidamente, hay que elegir entre practicar torpedos o no, y una vez tomada la decisión, comienza el juego. Este dispone de tres pantallas diferentes: una con la sala de máquinas, que es el lugar donde se encuentran todos los instrumentos necesarios para la navegación, otra con la sala de mapas, donde se tiene toda la información referente a la situación estratégica y, por último, la correspondiente al periscopio donde se encuentran todos los elementos necesarios para entrar en combate.

Si navegamos por la superficie o cerca de ésta, la aviación enemiga nos localizará enseguida y nos arrojará sus cargas de profundidad.

Una forma de evitar estas cargas, es navegar a gran profundidad. Sin embargo, existe el inconveniente de que al hacerlo, las baterías se irán descargando poco a poco, hasta llegar un momento que necesitemos subir a la superficie para cargarlas.

El ordenador nos avisa aproximadamente dos minutos y medio antes de que explote la carga, haciendo sonar una sirena. Si cuando esto ocurre estamos mirando por el periscopio, veremos atravesar la pantalla a los aviones.

El juego es bastante difícil de dominar, lo cual es un aliciente más para no aburrirnos. Los gráficos



están muy bien conseguidos, su desarrollo es bastante bueno, si bien la respuesta de las techas al pulsarlas no es todo lo rápida que se pudiera desear. En definitiva, es un buen juego para pasar el rato y al mismo tiempo aprender algo sobre navegación y combate.

Otro aliciente es el hecho de que se pueda jugar con dos Spectrum, conectados ambos a un Interface 1, y con sus propias pantallas de TV.

El tesoro de Adderonia

ATAQUE DE VIBORAS

Mogul/Software España

48 K

Tipo de juego: Arcade

PVP.: 1.650

El juego consiste en buscar un tesoro en el hogar de la diosa Adderonia. Hay que ir recogiendo las pepitas que se encuentran esparcidas por la pantalla, pero teniendo mucho cuidado de no ser alcanzados por la serpiente azul, nuestra principal enemiga. Además de ésta, hay otras atrapadas en varios lugares de la pantalla. Su número crecerá a medida que el juego vaya avanzando.

La víbora, al igual que



nosotros, puede ir recogiendo las pepitas, mientras que las otras no lo harán, aunque por supuesto lo que sí hacen es atacarnos.

Hay también esparcidos por la pantalla algunos diamantes que pueden hacer que aumente nuestra puntuación, y unos ópalos muy valiosos que va soltando la serpiente azul.

Es un juego entretenido que, aunque simple, resulta adicto para aquellos que se consideren hábiles. Está basado en el conocido Snake Pit.

■■■ *Durante las pasadas Navidades ha sido lanzado en Inglaterra **«Special Delivery»**, un juego sobre **Santa Claus**, por la compañía **Creative Sparks**. El juego está pensado para que sirva en cualquier época del año, ya que tiene cuatro pantallas diferentes.*

■■■ *Ya ha aparecido la primera persona capaz de llegar al final del juego de **Dinamic, BABALIBA.** Como prometieron los creadores, ha ganado un premio de 15.000 pts. por tal proeza.*

■■■ ***Sandman Coneth** ofrece un premio, consistente en un viaje a París, para la primera persona que logre completar su último juego, **Star Dreams.** El juego lo ha realizado la misma persona que creó el **Aural Quest** para el último LP del grupo **Stranglers.***

H	STOP in INPUT	1 ϕ	1

NUMERO DE SENTENCIA

NUMERO DE LINEA

SIGNIFICADO

CODIGO

Informes.

Ejemplo:

PRINT "hola":LOAD"":PRINT "FIN"

En este caso al pretender continuar una vez realizada la interrupción se nos presenta el mensaje:

ϕ OK

c) Cuando se interrumpe en la tercera o siguiente sentencias.
Ejemplo:

INPUT "> "; a : PRINT a : LOAD"": LETb =2

aparece el mensaje:

N Statement lost

al intentar continuar con la ejecución una vez interrumpido en la sentencia «LOAD».

Informes de pantalla

Para comunicarnos el ordenador el resultado de la ejecución de los programas, tanto si han sido completados, interrumpidos, o con errores, utiliza este las dos líneas inferiores de la pantalla para enviarnos dichos informes.

Estos informes son visualizados de acuerdo a un formato.

CODIGO: Número comprendido entre «ϕ» y «9» o letra de la «A» a la «R». Facilita la búsqueda en la tabla general de informes, apéndice B del manual (pág. 189).

SIGNIFICADO: Breve descripción del evento ocurrido. Para una mayor aclaración debe consultarse el manual.

LINEA: Número de línea donde se ha producido el evento.

SENTENCIA: Número de sentencia dentro de la línea.

Ejemplo:

Veamos el significado del informe:

H STOP in INPUT, 1ϕ : 1

«H» es el código y significa que ha ocurrido una interrupción del programa en la sentencia primera de la línea 1ϕ; dicha interrupción ha sido debida a que se ha introducido «STOP» en una sentencia del tipo «INPUT».

Un informe con número de sentencia «2» se refiere a la sentencia situada a continuación del primer separador (:) o de la palabra clave «THEN», y así sucesivamente.

Los comandos directos al no poseer número de línea, se indican en los informes como línea ϕ.


RUN

Acceso al teclado







MODO K VERIFY

Definición

«RUN» se utiliza normalmente como comando directo y permite al usuario, mediante su ejemplo, ejecutar un programa editado en lenguaje BASIC.

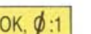
La estructura general de este comando es:

SENTENCIA	ARGUMENTO
RUN	N.º de línea

Ejemplos:
- RUN 3ϕ
- RUN 1ϕ
- RUN

Si no se especifica número de línea, el intérprete BASIC tomará, por defecto, como primera línea a ejecutar, la de numeración más baja.

Cuando el número de línea especificado en el argumento no exista, la ejecución del programa comenzará en la línea siguiente; si esta tampoco existiera, por que se encuentra fuera de la zona de nuestro programa BASIC, no se ejecutará el programa y además aparecerá el mensaje:


ϕ OK, ϕ:1

Edite el siguiente programa:

```
 1 REM ************
       *          *
       * PRUEBA   *
       *          *
       ************
10 PRINT "Linea 10"
20 PRINT "Linea 20"
30 PRINT "Linea 30"
40 PRINT "Linea 40"
50 PRINT "Linea 50"
60 PRINT "Linea 60"
```

Ejecútelo de las siguientes formas y compare los resultados:

- RUN
- RUN 1ϕ
- RUN 2ϕ
- RUN 3ϕ

———
———
———
———

- RUN 7ϕ

compare también los resultados proporcionados por los siguientes comandos directos:
- RUN 35
- RUN 4ϕ

Si en el argumento se especifica un número de línea comprendido entre "32768" y "61439" aparece el mensaje de error:

N Statement lost, ϕ:255

si es mayor a este último valor (61439) aparece:

B Integer out of range, ϕ:1

Una de las particularidades de la sentencia «RUN» es que borra la pantalla antes de ejecutar el programa almacenado en memoria.

Ejecute varias veces seguidas el siguiente programa y observe el efecto:

```
10 FOR n=1 TO 70
20 PRINT "randomize "
30 NEXT n
```

Otra de las particularidades, es que borra todas las variables que hasta ese momento estuvieran definidas.

Ejemplo:

— Edite estas dos líneas:

```
1ϕ PRINT a
2ϕ PRINT b
```

— Agisne unos valores a las variables «a» y «b» con comandos directos, por ejemplo:

```
LET a = 2ϕ
LET b = 127
```

— Compruebe los contenidos de dichas variables con:

```
PRINT a
PRINT b
```

— Ejecute el programa con «RUN», observará que en esta ocasión aparece el mensaje:

2 Variable not found, 1ϕ:1

ya que al ejecutarse «RUN» se han borrado las variables «a» y «b».

La sentencia «RUN» también puede ser incluida como línea dentro de un programa.

```
10 REM ************
       *          *
       * SIN FIN  *
       *          *
       ************
 20 PRINT "Este programa solo
puede"
 30 PRINT "pararse, pulsando "
 10 PRINT "CAPS SHIFT + SPACE"
 50 RUN
```

```
10 REM **************
       *            *
       * SIN FIN 2  *
       *            *
       **************
 20 INPUT "Cual es tu nombre?"
,n$
 30 PRINT "Te llamas ";n$
 40 PRINT #0;"Pulsa una tecla
para comenzar "
 50 PAUSE 0
 60 RUN
```

en estos dos ejemplos, el programa comienza a ejecutarse de nuevo al llegar a la última sentencia, de esta forma se crea un bucle sin fin.

El argumento también puede ser una variable numérica previamente definida.

Ejemplo:



```
  1 REM ***************
        *             *
        * RUN linea   *
        *             *
        ***************
  2 INPUT "En que linea quieres
 empezar la ejecucion del progra
ma? ",linea
  4 RUN linea
 10 PRINT "Ejecuto la 10"
 20 PRINT "Ejecuto la 20"
 30 PRINT "Ejecuto la 30"
 40 PRINT "Ejecuto la 40"
 50 PRINT "Ejecuto la 50"
 60 PRINT "Ejecuto la 60"
```

Una vez ejecutado el programa, éste pide que le introduzcamos el n.º de línea de la nueva ejecución, valor asignado a la variable «línea», la sentencia «RUN línea» lo hace a partir de este valor.

BREAK

Acceso al teclado



+

BREAK
SPACE

Definición

La función «BREAK» provoca una ruptura en la ejecución de un programa, en el acceso a los periféricos «impresora» y «cassette» y en los listados de más de 22 líneas, es decir, en

INICIO → LECTURA 1.ª INSTRUCCION → ANALISIS Y EJECUCION → CAPS SHIFT + SPACE ? — SI → BREAK; NO → ULTIMA INSTRUCCION ? — SI → FIN; NO → LECTURA SIGUIENTE INSTRUCCION → ANALISIS Y EJECUCION

INFORME
L BREAK into program

INFORME
Ø OK

Análisis sentencia «Break».

aquellos en que aparece el mensaje:

scroll?

Esta ruptura sólo provoca una *interrupción* en la ejecución del programa, es decir, que *no borra* el contenido de la memoria. En la mayoría de los casos se podrá continuar con ella, utilizando el comando «CONTINUE» (CONT).

ADVERTENCIA

Si se está ejecutando una sentencia del tipo «INPUT», no se puede provocar la ruptura del programa, esto se consigue utilizando otra técnica que posteriormente será descrita.

El intérprete BASIC al *terminar* de ejecutar una instrucción verifica si están pulsadas las teclas «CAPS SHIFT» y «SPACE», si no lo están continúa con la ejecución de la siguiente instrucción, y si por el contrario, lo están, provoca su interrupción.

En aquellas instrucciones en que el tiempo de ejecución es prolongado, es necesario *mantener* estas teclas oprimidas hasta que aparezca el informe correspondiente.

Ejemplo:

```
1Ø BEEP 2, 1Ø
2Ø BEEP 3, 2Ø
3Ø BEEP 1, 15
4Ø BEEP 3, 5
```


TIPO DE "INPUT"	METODO	
NUMERICO	STOP	
ALFANUMERICO	ELIMINAR	DELETE
	COMILLAS	EDIT
	+ STOP	
LINE	⇩ CAPS SHIFT + 6	

Ruptura en «Input».

«CONTINUE» repite la ejecución en la misma línea donde se provocó la interrupción.

Ejemplo:

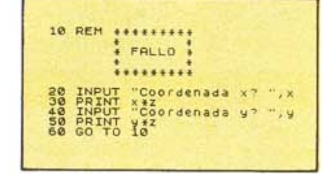
Introduzca las siguientes líneas

```
1Ø LOAD ""
2Ø PRINT "cargado"
```

cuando está en la fase de «carga» pulse la tecla «SPACE». Una vez interrumpido teclee «CONT», el programa volverá a ejecutar la sentencia «LOAD».

Si el programa se ha interrumpido debido a un error, podemos subsanar el problema momentáneamente y continuar con la ejecución.

Ejemplo:



```
10 REM ***********
       *         *
       * FALLO   *
       *         *
       ***********
20 INPUT "Coordenada x? ",x
30 PRINT x*z
40 INPUT "Coordenada y? ",y
50 PRINT y*z
60 GO TO 10
```

Este programa una vez ejecutado e introducido el valor de la «coordenada X» presenta un fallo en la línea 3Ø, ya que no está definida la variable «Z». Defínala con un comando directo como por ejemplo:

LET Z = 5

y tecleando «CONTINUE» volverá a ejecutarse el programa a partir de la línea 3Ø

«CONTINUE» *no se puede* emplear con comandos directos. Se pueden distinguir tres casos:

a) Cuando se interrumpe en la primera sentencia.

Ejemplo:

LOAD "" : LET a = 2 : PRINT a

si se pulsa «SPACE» (BREAK) y se pretende continuar con la ejecución de las restantes sentencias, el programa pierde el control y se queda un bucle sin fin, para salir de él pulse la tecla «CAPS SHIFT» + «SPACE».

b) Cuando se interrumpe en la segunda sentencia.

Para borrar las comillas, existen dos métodos. El primero es utilizando la función «DELETE», de esta forma desaparece la *comilla* situada a la izquierda del cursor. A partir de este instante ya se puede introducir la sentencia «STOP»; el mensaje que aparece es también:

H STOP in INPUT

En el segundo método se utiliza la función «EDIT», de esta manera desaparecen ambas comillas y al igual que en el caso anterior, se puede introducir «STOP».

Para cortar un «INPUT LINE», la filosofía es totalmente distinta, ya que ni se permite introducir «STOP», ni eliminar las comillas, simplemente por que estas no existen.

Ejemplo:

```
1∅ INPUT "  "; LINE a$
d∅ PRINT a$
3∅ GO TO 1∅
```

La única manera de salir del programa anterior, una vez ejecutado, es utilizando el cursor de desplazamiento inferior («CAPS SHIFT» + «6»). El mensaje presentado, al igual que en las anteriores situaciones es:

H STOP in INPUT

CONTINUE

Acceso al teclado

L PRINT



MODO K

PAPER

```
100 PRINT AT 3,0;"Este programa
se autodestruira"'"" al llegar
la cuenta a 00:00"
110 REM
          ***************
          *             *
          * CUENTA ATRAS *
          *             *
          ***************
120 FOR m=9 TO 0 STEP -1
130 PRINT AT 12,13;"0";m
140 FOR s=59 TO 0 STEP -1
142 IF s<10 THEN PRINT AT 12,16
;0;s: GO TO 155
150 PRINT AT 12,16;s
155 BEEP 0.02,20
160 NEXT s
170 NEXT m
180 REM
          ***************
          *             *
          * !!! BOOM !!! *
          *             *
          ***************
190 FOR n=0 TO 7
200 BORDER n
210 BEEP 0.03,n
220 NEXT n
230 CLS
235 FOR n=1 TO 50
240 PRINT OVER 1;AT 10,8;"!!!!
BOOM !!!!"
245 NEXT n
250 REM
           **************
           *            *
           * !!! NEW !!! *
           *            *
           **************
260 NEW
```

Definición

El comando directo «CONTINUE» se reconoce en el teclado por su forma abreviada «CONT». La utilidad de este comando es continuar con la ejecución de un programa que, debido a un informe de error o a un «BREAK» se ha interrumpido. Este comando no precisa argumento.

Cuando la interrupción se ha debido a un «BREAK» con informe:

L BREAK into program

o a una sentencia «STOP»:

9 STOP statement

el comando «CONTINUE» comienza a ejecutar el programa a partir de la siguiente instrucción, en que se produjo la ruptura.

Sin embargo, cuando el informe presentado por «BREAK» es:

D BREAK — CONT repeats

o se visualiza:

H STOP in INPUT

Ejecute estas sentencias y utilice la función «BREAK».

Dependiendo de la situación en que se utilice «BREAK», existen dos tipos de informes. Cuando se utiliza para interrumpir un programa, el informe visualizado en pantalla es:

L BREAK into program

En los restantes casos, con sólo mantener pulsada la tecla «SPACE» (BREAK) se consigue la interrupción, y el mensaje presentado por el ordenador es:

D BREAK — CONT repeats

La diferencia entre estos dos mensajes será explicada con detalle al tratar la sentencia «CONTINUE» (CONT).

Al final del informe aparece la línea y el número de sentencia, dentro de la línea donde se produjo la interrupción.

STOP

Acceso al teclado

SYMBOL SHIFT

READ

A STOP NEW

Tipo de sentencia

Comando de programación.

INICIO
CALCULO "1"
I/O
DECISION "1"? NO / SI
CALCULO "2"
DECISION "2"? SI / NO
STOP

INFORME
9 STOP statement

Ejemplo sentencia «Stop».

Definición

A pesar de ser un comando de programación, «STOP» es tratado en este capítulo, ya que su función es de control, cuando se ejecuta se interrumpe el programa.

«STOP» no tiene sentido como comando directo, por lo que debe ser editado como línea de programa *sin argumento*.

Ejemplo:

4∅∅ STOP

cuando el intérprete BASIC analiza esta sentencia, se detiene en la línea 4∅∅ y presenta el mensaje:

9 STOP statement, 4∅∅:1

Esta sentencia puede ser utilizada en diversas ocasiones, pero principalmente:

a) Para separar diversas rutinas independientes dentro de un solo programa.
Ejemplo:
Edite el programa «1» que simula una calculadora básica.

Las sentencias «STOP» separan las rutinas de «suma», «resta», «multiplicación» y «división». Para acceder a las diversas rutinas se utiliza la sentencia «RUN» y como argumento la variable «código», el valor de esta se asigna con el «INPUT» de la línea «8».

b) Para separar un programa principal de las *subrutinas*. Estas se estudiarán en otro capítulo.

c) Cuando se desea que se interrumpa la ejecución de un programa en función del resultado de una comparación. Se utiliza conjuntamente con el par de sentencias «IF-THEN».

Ejemplo:

```
10 REM ***********
       * COMPARA *
       ***********
20 INPUT "Numero 1? ";a1
30 PRINT a1
40 INPUT "Numero 2? ";a2
50 PRINT a2
60 IF a1=a2 THEN STOP
70 GO TO 10
```

se producirá la interrupción del programa, cuando las variables «al» y «a2» sean iguales.

d) En técnicas especiales de depuración de programas.

e) También se utiliza para provocar la ruptura de un programa, en una sentencia del tipo «INPUT».

Ruptura del «INPUT»

Para interrumpir la ejecución de un programa en una sentencia «INPUT», es necesario utilizar una serie de técnicas, dependiendo estas del tipo de «INPUT»:

— INPUT numérico.
— INPUT de cadena.
— INPUT LINE.

PROGRAMA 1

```
  10 REM **************
         *            *
         * CURSO BASIC *
         *            *
         **************
         *            *
         * CALCULADORA *
         *            *
         **************
  20 BORDER 4: PAPER 4: INK 1: C
LS
  30 REM ***********
         *         *
         * OPCIONES *
         *         *
         ***********
  40 PRINT AT 3,10;"CALCULADORA"
  50 PRINT AT 7,4;"CODIGO      OPE
RACION"
  60 PRINT AT 8,4;"______     ____
"______________
  70 PRINT AT 10,6;"14         SUMA
"
  80 PRINT AT 12,6;"22         REST
A"
  90 PRINT AT 14,6;"30         MULT
IPLICACION"
 100 PRINT AT 16,6;"38         DIVI
SION"
 110 INPUT "Introduzca codigo de
operacion  > ";codigo
 120 CLS
 125 RUN codigo*10
 130 REM *******
         *     *
         * SUMA *
         *     *
         *******
 140 PRINT AT 3,13;"SUMA"
 150 INPUT "Sumando 1? ";suma1
 160 INPUT "Sumando 2? ";suma2
 170 LET suma=suma1+suma2
 180 CLS
 190 PRINT suma1;" + ";suma2;" =
";suma
 200 STOP
 210 REM ********
         *      *
         * RESTA *
         *      *
         ********
 220 PRINT AT 3,12;"RESTA"
 230 INPUT "Minuendo? ";min
 240 INPUT "Sustraendo? ";sus
 250 LET resta=min-sus
 260 CLS
 270 PRINT min;" - ";sus;" = ";r
esta
 280 STOP
 290 REM ****************
         *              *
         * MULTIPLICACION *
         *              *
         ****************
 300 PRINT AT 3,8;"MULTIPLICACIO
N"
 310 INPUT "Multiplicando? ";mul
 320 INPUT "Multiplicador? ";mul
t
 330 LET multiplicacion=mul*mult
 340 CLS
 350 PRINT mul;" * ";mult;" = ";
multiplicacion
 360 STOP
 370 REM ***********
         *         *
         * DIVISION *
         *         *
         ***********
 380 PRINT AT 3,12;"DIVISION"
 390 INPUT "Dividendo? ";div
 400 INPUT "Divisor? ";divi
 410 LET division=div/divi
 420 CLS
 430 PRINT div;" / ";divi;" = ";
division
 440 STOP
```

PROGRAMA 2

```
  10 REM **************
         *            *
         * CURSO BASIC *
         *            *
         **************
         *            *
         * !!! NEW !!! *
         *            *
         **************
  20 BORDER 6: PAPER 6: INK 1: C
LS
  30 REM ***************
         *             *
         * DIBUJO CRONO *
         *             *
         ***************
  40 PRINT AT 10,11; PAPER 2; IN
K 4; FLASH 1;"▚▚▚▚▚▚▚▚▚"
  50 FOR n=11 TO 13
  60 PRINT AT n,11; PAPER 2; INK
 4; FLASH 1;"▚";AT n,19;"▚"
  70 NEXT n
  80 PRINT AT 14,11; PAPER 2; IN
K 4; FLASH 1;"▚▚▚▚▚▚▚▚▚"
  90 PRINT AT 12,13;"10:00"
```

Cuando son del tipo numérico basta simplemente con teclear la sentencia «STOP» (SYMBOL SHIFT + A) y «ENTER», inmediatamente se provoca la ruptura del programa y aparece el mensaje:

```
H STOP in INPUT
```

Ejemplo:

```
1Ø INPUT "> ";a
2Ø PRINT a
3Ø GO TO 1Ø
```

Si intenta utilizar la función «BREAK», observará que no sirve en estos casos.

Puede obtener la ruptura también, de una forma menos elegante, tecleando letras aleatoriamente, de esta manera el intérprete BASIC al analizar la entrada de datos y comprobar que no corresponde con un valor numérico o con una variable previamente definida, visualizará el mensaje:

```
2 Variable not found
```

Cuando el «INPUT» es del tipo alfanumérico, la técnica es ligeramente distinta.

Ejemplo:

```
1Ø INPUT "? "; a$
2Ø PRINT a$
3Ø GO TO 1Ø
```

Intente introducir «STOP», observará que el programa no se interrumpe, ya que la variable alfanumérica «a$» asume el código correspondiente al *token* «STOP», y se ejecuta la siguiente instrucción, y así sucesivamente. La única manera de introducir «STOP», sin que lo tome como valor alfanumérico, es eliminando las comillas ("").

MEMORIOM

Mario Fidel RODRIGUEZ

Spectrum 48 K

Premiado con 15.000 ptas.

Se trata éste de un juego entretenido que puede recordarnos, levemente, el antiguo y conocido juego de las familias, pero simplificado.

Como podrá leer en las instrucciones que dan paso al juego, éste no consiste en otra cosa más que en utilizar, de la mejor manera, nuestra memoria y recordar la carta que puede emparejar con la última sacada, según vayan apareciendo en la pantalla. Como es lógico, ganará aquel jugador que más parejas logre reunir.

Cada jugador podrá levantar carta dos veces y, si consigue formar una pareja, tendrá otra oportunidad.



A. PERERA

```
1000 REM pantalla inicial
1010 CLS : PRINT AT 1,3;"* * *
 MEMORION    * * *": PRINT AT 5,0
;" I = instrucciones": PRINT AT
7,0;" J = iniciar juego": FLASH
1: PRINT AT 19,0;"© MARIO FIDEL"
: FLASH 0
1020 IF INKEY$="" THEN GO TO 102
0
1025 IF INKEY$="I" OR INKEY$="i"
 THEN BEEP 0.1,10: GO TO 1050
1030 IF INKEY$="J" OR INKEY$="j"
 THEN BEEP 0.1,10: GO TO 2000
1040 BEEP 0.5,-30: GO TO 1020
1050 CLS : RESTORE : FOR i=0 TO
16: READ m$: FOR j=1 TO LEN m$:
PRINT AT i,j-1;m$(j): BEEP 0.01,
I+J+10: NEXT j: NEXT i
1055 BRIGHT 1: PRINT AT 19,0;"
    TECLEA PARA EMPEZAR          ":
 BRIGHT 0
1060 IF INKEY$="" THEN GO TO 106
0
1070 BEEP 0.1,10: GO TO 2000
1200 DATA "          INSTRUCCIONES
"
1201 DATA " "
1202 DATA "EL JUEGO CONSISTE EN
USAR TU"
1203 DATA "MEMORIA, BUSCANDO CAR
TAS COMO"
1204 DATA "LAS QUE VAYAN APARECI
ENDO EN LA"
1205 DATA "PANTALLA."
1206 DATA " "
1207 DATA "CADA JUGADOR PODRA LE
VANTAR DOS"
1208 DATA "VECES. SI CONSIGUE HA
LLAR DOS"
1209 DATA "IGUALES, VUELVE A JUG
AR..."
1210 DATA " "
1211 DATA "PARA SELECCIONAR UNA
CARTA."
1212 DATA "TECLEAR SUS COORDENAD
AS, CON"
1213 DATA "UNA LETRA Y UN NUMERO
 (A 2)"
1214 DATA " "
1215 DATA "GANARA QUIEN MAS PARE
JAS HAGA."
1216 DATA " "
2000 REM seleccion de jugadores
2010 CLS : PRINT AT 1,0;"CUANTA
GENTE VAIS A JUGAR ???";';"(MAS
DE UNO Y MENOS DE SEIS)": INPUT
AT 3,0;NJ: IF NJ<2 OR NJ>5 THEN
GO TO 2010
2012 BEEP 0.1,10: PRINT AT 3,0;N
J
2020 DIM N$(5,8): DIM P(5)
2021 LET P(1)=32
2030 FOR I=1 TO NJ: BEEP 0.2,0:
PRINT AT 3+2*I,0;"COMO TE LLAMAS
, JUGADOR ";I;" ???": INPUT N$(I
): LET P(I)=0: BRIGHT 1: PRINT A
T 4+2*I,0;N$(I): BRIGHT 0: NEXT
I
2032 PRINT AT 16,0;"NIVEL DE DIF
ICULTAD? (1-2-3)": PRINT AT 17,0
;"1=FACIL, 2=NORMAL, 3=DIFICIL)"
2033 INPUT lim: LET lim=lim*2+2
2034 IF lim=4 THEN LET v$="F A C
 I L"
2035 IF lim=6 THEN LET v$="N O R
 M A L"
2036 IF lim=8 THEN LET v$="D I F
 I C I L"
2037 BRIGHT 1: PRINT AT 18,0;v$:
 BRIGHT 0
2040 DIM c(8,8): DIM c$(8,8,1)
2100 REM colocacion de caractere
s
2105 BEEP 0.2,0: BEEP 0.2,4: BEE
P 0.2,7: CLS : FLASH 1: PRINT AT
 9,0;"UN MOMENTO, QUE ESTOY PREP
ARANDO";"LO NECESARIO PARA JUGAR
...      ": FLASH 0
2110 FOR i=1 TO lim: FOR j=1 TO
lim: LET c(i,j)=0: LET c$(i,j)="
": NEXT j: NEXT i
2116 IF lim=4 THEN LET j$="ZEACL
UAZEUOICLOI"
2118 IF lim=6 THEN LET j$="PQKAJ
SOWIHDGFEURYTQPWOEIRUTYASDFGHJK"
2120 IF LIM=8 THEN LET j$="QPIOW
3UEYRTAZS5XD4CFVGB2HNJ1MK0L5QAZW
SXEDC4RFVT23GBYHNUJMI1KOLP0"
2130 FOR i=1 TO lim*lim
2140 LET k=INT (lim*RND)+1: RAND
OMIZE : LET l=INT (lim*RND)+1: I
F c$(k,l)=" " THEN GO TO 2200
2150 REM buscar sitio libre
2159 LET d$="."
2160 FOR m=1 TO lim: FOR n=1 TO
lim: IF c$(m,n)=" " AND d$<>j$(i
) THEN LET k=m: LET l=n: GO TO 2
200
2165 LET d$=c$(m,n)
2170 NEXT n: NEXT m
2200 LET c$(k,l)=j$(i)
2210 NEXT i
3000 REM pantalla del juego
3010 CLS : FOR i=4 TO lim*16+4 S
TEP 16
3020 PLOT i,44+16*(8-lim): DRAW
0,lim*16
3030 NEXT i
3040 FOR i=44+16*(8-lim) TO 16*l
im+44+16*(8-lim): STEP 16
3050 PLOT 4,i: DRAW lim*16,0
3060 NEXT i: BRIGHT 1
3062 IF lim=4 THEN LET x$="A B C
 D": LET xx=9
3063 IF lim=6 THEN LET x$="A B C
 D E F": LET xx=13
3064 IF lim=8 THEN LET x$="A B C
 D E F G H": LET xx=17
3070 FOR i=1 TO lim: FOR j=1 TO
lim: PRINT AT i*2-1,j*2-1;" ": N
EXT j: PRINT AT i*2-1,xx;i: NEXT
 i: PRINT AT xx,1;x$: BRIGHT 0
3080 FOR i=1 TO nj
3090 PRINT AT 1+i*2,19;i;" ";n$(
i)
3100 NEXT i
4000 REM cuerpo del juego
4001 LET I=0: DIM A$(2): DIM A(2
): DIM B(2): DIM f(2): DIM g(2)
4010 LET I=I+1
4020 IF I>nj THEN LET I=1
4025 BRIGHT 1: PRINT AT 1,21;"LE
 TOCA A:": BRIGHT 0: PAUSE 50
4030 FLASH 1: BEEP 0.2,10: PRINT
 AT 1+I*2,21;n$(I);" ";p(i): FLA
SH 0
4035 FOR o=1 TO 2
4040 GO SUB 4500
4046 IF c(y1,x1)<>0 THEN BEEP 0.
5,-30: GO TO 4040
4050 LET a$(o)=c$(y1,x1): LET a(
o)=k: LET b(o)=l: LET f(o)=y1: L
ET g(o)=x1
4055 IF o=2 AND a(1)=a(2) AND b(
=lim*lim/2 THEN GO TO 9000
4105 PRINT AT 1+i*2,21;n$(i);" "
;p(i)
4110 IF z=1 THEN FLASH 1: PRINT
AT 19,0;"******* JUEGAS DE NUEVO
 *******": FLASH 0: FOR u=1 TO 5
: BEEP 0.05,20: BEEP 0.05,23: BE
EP 0.05,25: NEXT u: PAUSE 100: P
RINT AT 19,0;"
               ": GO TO 4030
4120 GO TO 4010
4500 REM seleccion de carta
4505 IF INKEY$<>"" THEN GO TO 45
05
4510 BRIGHT 1: PRINT AT 19,0;"te
clea letra-numero _ _": BRIGHT 0
4520 IF INKEY$="" THEN GO TO 452
```

```
0
4530 LET l$=INKEY$: IF lim=8 AND
 (l$<"a" OR l$>"h") OR lim=6 AND
 (l$<"a" OR l$>"f") OR lim=4 AND
 (l$<"a" OR l$>"d") THEN BEEP 0.
1)=b(2) THEN BEEP 0.5,-30: GO TO
 4040
4060 IF o=1 THEN BEEP 0.1,15: FL
ASH 1: PRINT AT 1,21;"JUEGA OTRA
": FLASH 0
4070 NEXT o
4080 IF a$(1)=a$(2) THEN LET i1=
f(1): LET j1=g(1): LET i2=f(2):
LET j2=g(2): LET c(i1,j1)=1: LET
 c(i2,j2)=1: LET p(i)=p(i)+1: PR
INT AT 1+i*2,30;p(i): BEEP 0.5,1
3: BEEP 0.5,17: BEEP 0.5,20: LET
 z=1: PRINT AT a(1),b(1);c$(i1,j
1): PRINT AT a(2),b(2);c$(i2,j2)
: IF p(1)+p(2)+p(3)+p(4)+p(5)<li
m*lim/2 THEN GO TO 4110
4085 IF p(1)+p(2)+p(3)+p(4)+p(5)
=lim*lim/2 THEN GO TO 9000
4090 LET z=0: BEEP 1,-30: FOR o=
1 TO 2: BRIGHT 1: PRINT AT a(o),
b(o);" ": BRIGHT 0: NEXT o
4100 IF p(1)+p(2)+p(3)+p(4)+p(5)
2,-30: GO TO 4520
4531 PRINT AT 19,20;l$: BEEP 0.0
3,30
```

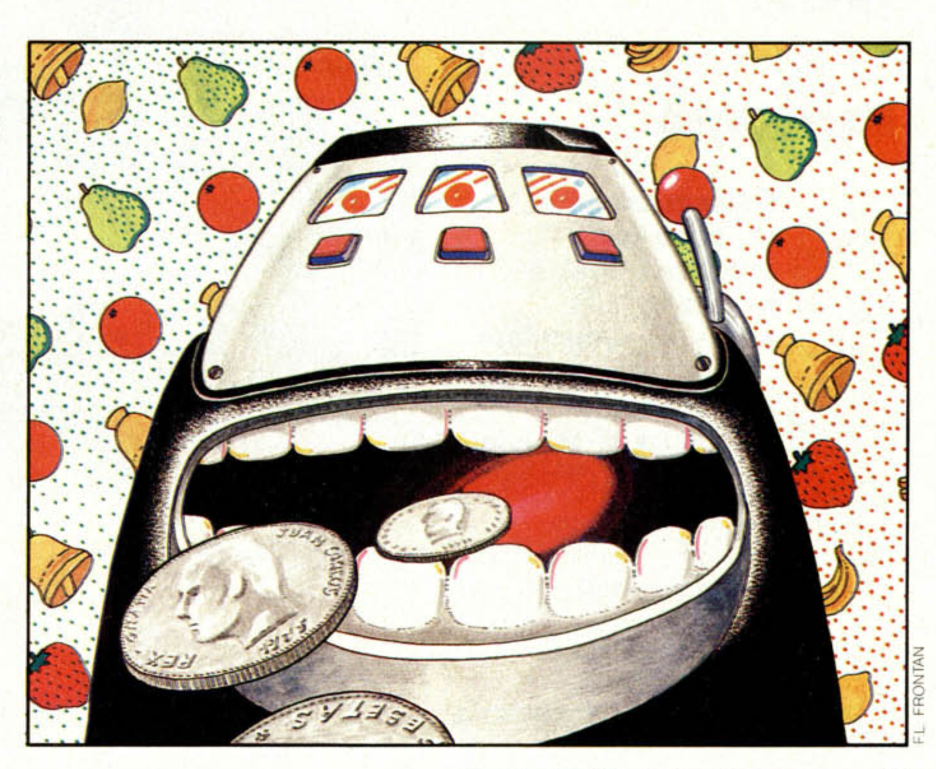
```
4535 IF INKEY$<>"" THEN GO TO 45
35
4540 IF INKEY$="" THEN GO TO 454
0
4543 LET q$=INKEY$: IF lim=8 AND
 (q$<"1" OR q$>"8") OR lim=6 AND
 (q$<"1" OR q$>"6") OR lim=4 AND
 (q$<"1" OR q$>"4") THEN BEEP 0.
2,-30: GO TO 4535
4545 LET nu=0
4550 LET nu=VAL (q$)
4551 PRINT AT 19,22;nu: BEEP 0.0
3,30
4555 IF INKEY$<>"" THEN GO TO 45
55
4560 LET x1=CODE l$-96: LET y1=n
u
4561 FLASH 1: IF c(y1,x1)<>0 THE
N FLASH 0
4570 LET k=y1*2-1: LET l=x1*2-1:
 PRINT AT k,l;c$(y1,x1): FLASH 0
4580 RETURN
9000 REM final del juego
9001 CLS : FLASH 1: PRINT AT 0,3
;"N I V E L     ";V$: FLASH 0: P
RINT AT 2,7;"JUGADOR     PUNTOS"
: PRINT AT 3,7;"----------- ----
--"
9002 LET MAY=0
9010 FOR i=1 TO nj
```

```
9020 PRINT AT I*2+3,7;I;"   ";N$(
I);"   ";P(I)
9022 IF P(I)>MAY THEN LET camp=I
: LET may=p(i)
9030 NEXT I
9040 PRINT AT 16,0;"EL CAMPEON
ES ";N$(camp): PRINT AT 17,0;"HA
 DESCUBIERTO ";P(camp);" PAREJAS
!!!"
9050 LET Y$="E R E S   U N    M
E M O R I O N": FOR I=1 TO LEN Y
$: BEEP 0.05,I*2-10: PRINT AT 19
,I-1;Y$(I): NEXT I
9060 RESTORE 9090: FOR i=1 TO 6:
 READ dur,not: BEEP dur,not: NEX
T i
9070 PAUSE 30
9080 FOR i=1 TO 7: READ dur,not:
 BEEP dur,not: NEXT i
9090 DATA 0.3,5,0.3,5,0.5,0,0.5,
5,0.5,2,0.6,2
9091 DATA 0.3,4,0.3,5,0.5,7,0.5,
4,0.5,5,0.15,9,0.4,5
9100 CLS : PRINT AT 11,0;"QUEREI
S JUGAR OTRA VEZ ??? (S-N)"
9110 IF INKEY$="" THEN GO TO 911
0
9120 IF INKEY$="S" OR INKEY$="s"
 THEN GO TO 2000
9130 STOP
```

MAQUINA TRAGAPERRAS

Bernardo LOPEZ GARCIA

Spectrum 16 K



F.L. FRONTAN

En esta ocasión, nuestro Spectrum se verá convertido en una auténtica máquina tragaperras que nos dará premios variados, según la combinación de cuatro frutas que aparecerán en pantalla.

El programa nos informará constantemente del dinero que disponemos, el número de la jugada y el premio que recibimos al lograr una combinación. Así pues, iniciamos el juego con 200 ptas. que irán disminuyendo en cada jugada si no obtenemos premio. Cuando lo obtenemos, oimos una música, y, al dejar de jugar, el ordenador nos informa del dinero que hemos ganado o perdido.

Para comenzar la partida, hemos de pulsar «j». El juego termina a las diez jugadas, al acabarse el dinero o cuando pulsamos «p».

El programa se carga con Load o Load «frutas» y, una vez cargado, se ejecuta automáticamente. ¡Suerte!

```
NOTAS GRAFICAS
--------------
A B C D E F G H I J K L M N
```

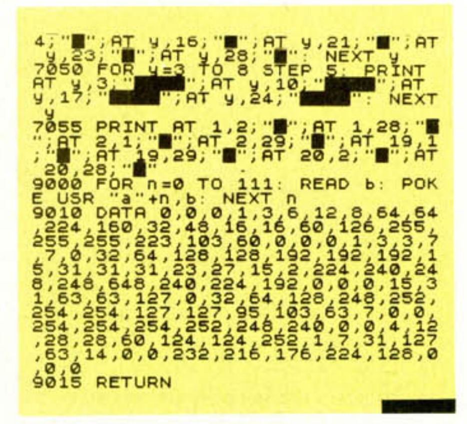
Premiado con 15.000 ptas.

```
   1 REM ** B.LOPEZ.GARCIA **
        *********************
   5 BORDER 0: PAPER 0: INK 6: C
LS
  10 GO SUB 7000
  20 LET p=200
  30 LET j=0: LET f=00
  40 LET a$=" ": LET e$=" "
  50 LET b$=" ": LET f$=" "
  60 LET c$=" ": LET g$=" "
  70 LET d$=" ": LET h$=" "
  80 PRINT AT 13,8;"FONDOS ";AT
13,15; FLASH 1; INK 2;p;"  "; FL
ASH 0;AT 13,20;"Pts"
 100 PRINT AT 15,8;"PREMIO ";AT
15,15; FLASH 1; INK 2;f;"  "; FL
ASH 0;AT 15,20;"Pts"
 110 PRINT AT 17,10;"JUGADA ";AT
 17,18; FLASH 1; INK 2;j
 120 IF p<25 THEN CLS : PRINT AT
 13,7; FLASH 1;"NO TIENES FONDOS
": STOP
 130 IF j>=10 THEN CLS : PRINT A
T 13,7; FLASH 1;"ES TARDE EL CAS
INO CIERRA";AT 15,7;"SE RETIRO C
ON ";p;" Ptas": STOP
 140 PAUSE 0
```

```
 150 IF INKEY$="j" OR INKEY$="J"
 THEN GO TO 175
 160 IF INKEY$="p" OR INKEY$="P"
 THEN CLS : PRINT AT 13,7; FLASH
 1;"ADIOS";AT 15,7;"SE RETIRO CO
N ";p;" Ptas": STOP
 170 IF INKEY$<>"p" AND INKEY$<>
"P" THEN GO TO 140
 175 LET j=j+1: LET p=p-25: LET
f=00
 180 LET y=5: LET w=6: LET x=4:
FOR v=1 TO 20
 190 GO SUB 6000
 195 LET u=s
 200 LET x=11: FOR v=1 TO 20
 210 GO SUB 6000
 215 LET d=s: LET x=18: FOR v=1
TO 20
 220 GO SUB 6000
 225 LET t=s: LET x=25: FOR v=1
TO 20
 230 GO SUB 6000
 235 LET q=s
 240 IF u=d AND d=t AND t=q THEN
 LET p=p+525: LET f=500: GO SUB
6500
 250 IF u=d AND d=t AND t<>q THE
```

```
N LET p=p+125: LET f=100: GO SUB
 6500
 260 IF u=d AND d<>t AND t=q THE
N LET p=p+100: LET f=75: GO SUB
6500
 270 IF u=t AND u<>d AND d=q THE
N LET p=p+125: LET f=100: GO SUB
 6500
 280 IF u<>d AND u=q AND d=t THE
N LET p=p+275: LET f=250: GO SUB
 6500
 285 IF u<>d AND d=t AND t=q THE
N LET p=p+125: LET f=100: GO SUB
 6500
 290 GO TO 80
6000 PRINT AT y,x; INK 2;a$;AT w
,x;e$;AT y,x; INK 4;b$;AT w,x;f$
;AT y,x; INK 2;c$;AT w,x;g$;AT y
,x; INK 6;d$;AT w,x;h$: BEEP .01
,40: NEXT v: LET s=INT (4*RND)
6100 IF s=0 THEN PRINT AT y,x; I
NK 2;a$;AT w,x;e$
6110 IF s=1 THEN PRINT AT y,x; I
NK 4;b$;AT w,x;f$
6120 IF s=2 THEN PRINT AT y,x; I
NK 2;c$;AT w,x;g$
```

```
6130 IF s=3 THEN PRINT AT y,x; I
NK 6;d$;AT w,x;h$
6135 RETURN
6520 BEEP .1,0: BEEP .1,3: BEEP
.1,2: BEEP .1,1: BEEP .3,0: BEEP
 .1,1: BEEP .1,7: BEEP .2,3: BEE
P .1,2: BEEP .3,4: BEEP .5,2: BE
EP .2,1: BEEP .2,7
6530 RETURN
7000 FOR x=3 TO 27
7010 PRINT AT 0,x;"■";AT 21,x;"■
": NEXT x
7015 FOR y=3 TO 18: FOR x=0 TO 3
0 STEP 30
7020 PRINT AT y,x;"■": NEXT x: N
EXT y
7025 FOR x=7 TO 23
7030 PRINT AT 11,x;"■";AT 19,x;"
■": NEXT x
7035 FOR y=12 TO 18: FOR x=6 TO
24 STEP 18
7040 PRINT AT y,x;"■": NEXT x: N
EXT y
7045 FOR y=4 TO 7: PRINT AT y,2;
"■";AT y,7;"■";AT y,9;"■";AT y,1
```



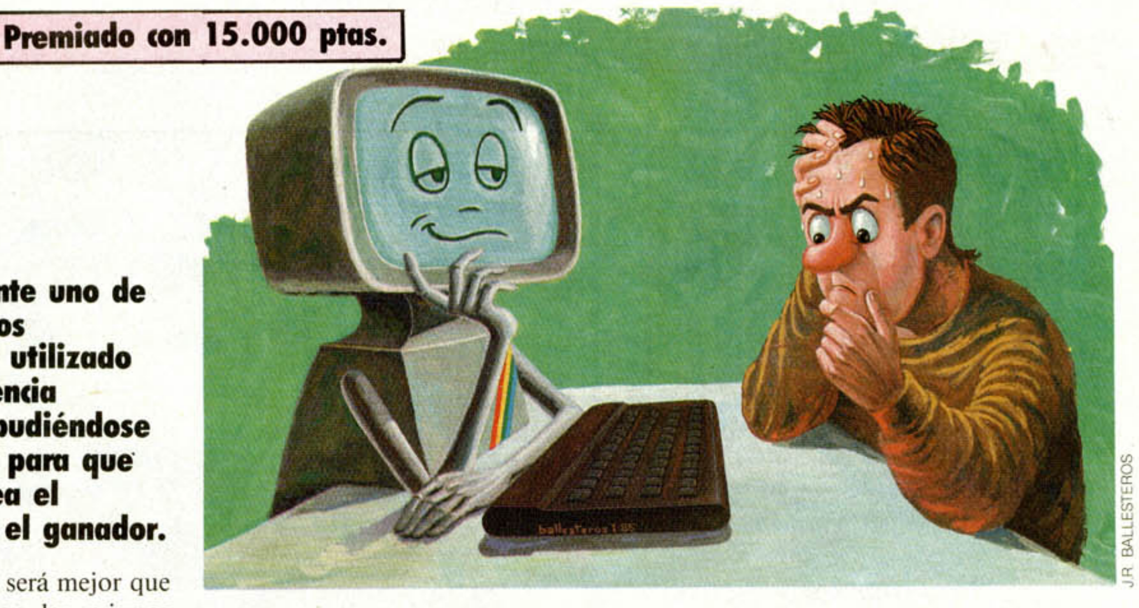
```
4;"■";AT y,16;"■";AT y,21;"■";AT
 y,23;"■";AT y,28;"■": NEXT y
7050 FOR y=3 TO 8 STEP 5: PRINT
AT y,3;"■■■■■";AT y,10;"■■■■■";AT
y,17;"■■■■■";AT y,24;"■■■■■": NEXT
 y
7055 PRINT AT 1,2;"■";AT 1,28;"■
";AT 2,1;"■";AT 2,29;"■";AT 19,1
;"■";AT 19,29;"■";AT 20,2;"■";AT
 20,28;"■"
9000 FOR n=0 TO 111: READ b: POK
E USR "a"+n,b: NEXT n
9010 DATA 0,0,0,1,3,6,12,8,64,64
,224,160,32,48,16,16,60,126,255,
255,255,223,103,60,0,0,0,1,3,3,7
,7,0,32,64,128,128,192,192,192,1
5,31,31,31,23,27,15,2,224,240,24
8,248,648,240,224,192,0,0,0,15,3
1,63,63,127,0,32,64,128,248,252,
254,254,127,127,95,103,63,7,0,0,
254,254,254,252,248,240,0,0,4,12
,28,28,60,124,124,252,1,7,31,127
,63,14,0,0,232,216,176,224,128,0
,0,0
9015 RETURN
```

NIM

Premiado con 15.000 ptas.

A. FROUFE

16 K



J.R. BALLESTEROS

Estamos ante uno de los primeros programas utilizado en inteligencia artificial, pudiéndose programar para que siempre sea el ordenador el ganador.

Ante esta perspectiva, será mejor que empecemos el juego con las mismas oportunidades ambos y procurar dar una buena paliza al contrincante, cosa que, como podrá apreciar, no es nada fácil.

A pesar de que las instrucciones de manejo están perfectamente especificadas en el listado, podemos decirle, generalizando, que el juego consiste en intentar retirar todas las fichas de las tres filas que aparecen en la pantalla, cada uno con un número determinado de ellas. El ganador, será el que consiga hacerlo antes.

```
NOTAS GRAFICAS
--------------
     A B
     ✱ X
```

```
  10 REM     NIM        ©1984
  20 REM  A.Froufe y MICRO-HOBBY
  30 FOR n=USR "a" TO USR "b"+7:
 READ y: POKE n,y: NEXT n
  40 DATA 0,90,60,126,102,60,90,
0
  50 DATA 129,66,36,24,24,36,66,
129
  55 LET i=0: LET j=0: LET b=1
  60 DIM a(3): DIM b(3,4): DIM s
(4): DIM z$(30)
  65 BORDER 0: PAPER 0: INK 7: C
LS
  70 PRINT PAPER 6;AT 0,0;z$;AT
0,11; PAPER 0;" N I M "
  80 PRINT : PRINT
  90 LET w$="   En pantalla apar
eceran tres     filas de piezas,
 cada una       con un numero de
terminado       de ellas.
"
 100 LET w$=w$+"   En cada tirad
a debes ele-      gir una fila
y el numero       de piezas que
 quieres re-      tirar de ella
.
"
 110 LET w$=w$+"   En cada jugad
a solo puedes     retirar ficha
s de una fila     y siempre ten
dras que re-      tirar al meno
s una ficha       de la fila el
egida.
"
 120 LET w$=w$+"   Ganara el que
 retire la        ultima ficha
del tablero."
 130 FOR x=1 TO LEN w$: IF w$(x)
=" " THEN PRINT " ";: GO TO 150
 140 PRINT w$(x);: BEEP .02,12
 150 NEXT x
 160 INPUT "Tu nombre, por favor
";n$
 170 IF LEN n$>9 THEN PRINT AT 2
1,0;"No puedes sobrepasar 9 letr
as": PAUSE 200: GO TO 160
 180 CLS : PRINT PAPER 6;AT 0,0;
z$+"  ";AT 0,12; PAPER 0;" N I M
"
 190 PRINT AT 20,0; INK 5;"XXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXX"
 200 FOR n=1 TO 20: PRINT INK 5;
AT n,0;"X";AT n,31;"X": NEXT n
 210 FOR p=1 TO 3: LET a(p)=INT
(10*RND)+6: NEXT p
 220 PRINT INK 2;AT 5,1;"filas";
AT 5,23;"cantidad"
 230 GO SUB 800
 260 PRINT AT 21,0;"FILA ?     "
 270 LET k$=INKEY$
 280 IF k$="" THEN GO TO 270
 290 IF CODE k$<49 OR CODE k$>51
 THEN GO TO 260
 300 BEEP .1,30: LET p=VAL k$: P
RINT AT 21,0;z$
 310 PRINT AT 21,0;"CANTIDAD ?"
 320 INPUT n: IF n<1 OR n>a(p) T
HEN GO TO 260
 330 PRINT AT 21,0;z$
 340 LET a(p)=a(p)-n
 350 IF a(1)+a(2)+a(3)>0 THEN GO
 SUB 800: GO TO 380
 360 PAPER 5: CLS : PRINT AT 11,
10;"TU GANAS": LET j=j+1: FOR p=
1 TO 250: NEXT p: PAPER 0
 370 GO TO 900
 380 FOR p=1 TO 3
 390 LET x=a(p)
 400 FOR m=4 TO 1 STEP -1
 410 LET y=INT (x/2)
 420 LET b(p,m)=x-2*y
 430 LET x=y
 440 NEXT m
 450 NEXT p
 460 FOR m=1 TO 4
 470 LET x=b(1,m)+b(2,m)+b(3,m)
 480 LET s(m)=x-2*INT (x/2)
 490 NEXT m
 500 IF s(1)+s(2)+s(3)+s(4)>0 TH
EN GO TO 560
510 FOR p=1 TO 3
520 IF a(p)=0 THEN GO TO 550
530 LET n=INT (a(p)*RND+1)
540 GO TO 710
550 NEXT p
560 FOR m=1 TO 4
570 IF s(m)>0 THEN GO TO 590
580 NEXT m
590 FOR p=1 TO 3
600 IF b(p,m)=1 THEN GO TO 620
610 NEXT p
620 LET n=0
630 FOR m=m TO 4
640 IF s(m)=0 THEN GO TO 700
650 LET x=2↑(4-m)
660 IF b(p,m)=0 THEN GO TO 690
670 LET n=n+x
680 GO TO 700
690 LET n=n-x
700 NEXT m
710 PRINT AT 21,0;"YO MUEVO ";n
;" DE LA FILA ";p: FOR a=1 TO 25
0: NEXT a
 720 PRINT AT 21,0;z$
 730 LET a(p)=a(p)-n
 740 IF a(1)+a(2)+a(3)>0 THEN GO
 TO 760
 750 LET i=i+1: CLS : PRINT AT 1
```

```
1,13;"YO GANO": FOR a=1 TO 250:
NEXT a: GO TO 900
 760 GO SUB 800
 770 GO TO 260
 800 FOR p=1 TO 3: PRINT AT 3+4*
p,6;"                          ": I
F a(p)=0 THEN GO TO 820
 810 PRINT AT 4*p+3,3;p;"  ";: F
OR g=1 TO a(p): PRINT INK 4; BRI
GHT 1;"*";: NEXT g: PRINT AT 4*p
+3,26;a(p)
 820 NEXT p
 830 PRINT AT 2,2;n$;" ";j: PRIN
T AT 2,20;"SPECTRUM ";i
 840 RETURN
 900 CLS : PRINT AT 20,4;"Jugamo
s de nuevo ? (s/n)"
 910 LET k$=INKEY$
 920 IF k$="s" THEN GO TO 180
 930 IF k$<>"s" AND k$<>"n" THEN
 GO TO 910
 940 CLS : GO TO 10000
```

¡ES LA GUERRA!

Pilar FERNANDEZ CANTON

Spectrum 16 K

La guerra ha sido una constante en el comportamiento humano, constante que, a unos espanta y, a otros «entretiene» siempre y cuando se trate de ficción.

Es este el caso del programa que presentamos a continuación, en el que tendremos que enfrentarnos al ataque constante de infinidad de tanques que intentan abatirnos. Nuestra misión será evitar esta maniobra enemiga y adelantarnos, disparando y destruyéndole.

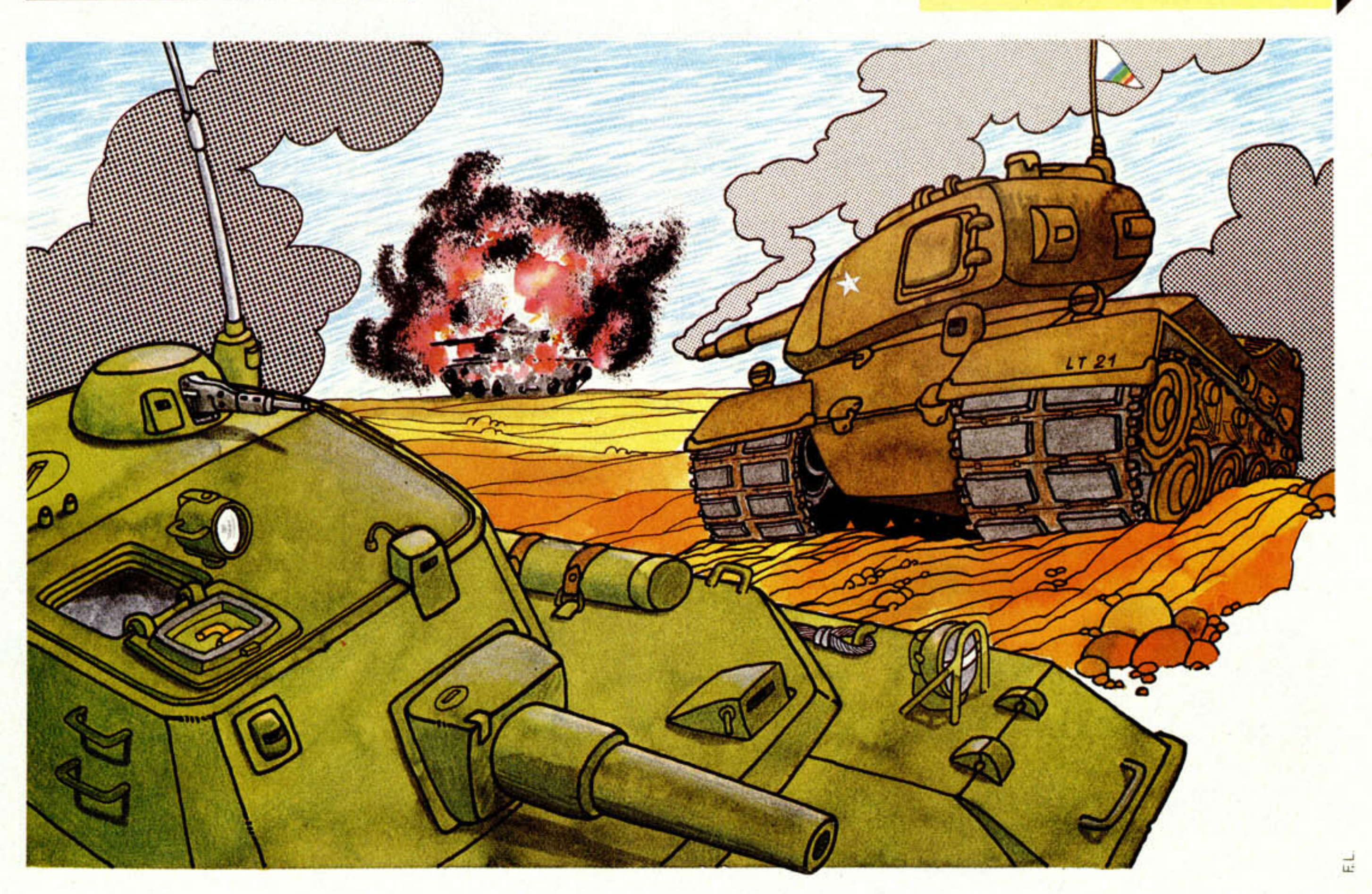
Este es, en síntesis, el objetivo del juego en el que nos veremos inmersos en plena batalla y donde tendremos que demostrar rapidez y estrategia para ganar la «guerra» y salvar nuestro pellejo.

Premiado con 15.000 ptas.

NOTAS GRAFICAS

A B C D E F G H I J K L M N O P Q R S T U

```
   1 REM TANQUES
   5 BORDER 1: PAPER 5: INK 2: C
LS : PRINT AT 10,3;"Espera unos
segundos"
  10 FOR w=USR "a" TO USR "a"+7
  20 READ a$
  25 LET a$=a$(3 TO )+"00000000"
+a$( TO 2)
  27 IF LEN a$<>24 THEN STOP
  30 FOR i=1 TO 4
  40 LET a$=a$(23 TO 24)+a$( TO
22)
  71 LET a1= VAL ("BIN "+a$( TO
8))
  75 LET a2=VAL ("BIN "+a$(9 TO
16))
  80 LET a3=VAL ("BIN "+a$(17 TO
 24))
  90 POKE w+24*i-24,a1: POKE w+2
4*i-16,a2: POKE w+24*i-8,a3
  93 NEXT i
  95 NEXT w
 100 REM "                    "
 200 DATA "0000000000000000","00
00000001100000","111111111111111
0","0000001111111110","000001111
1111100","0011111111111110","001
0000000000001","0001111111111110
"
 440 RESTORE 470
 450 FOR f=USR "m" TO USR "u"+7
```



```
 460 READ a: POKE f,a: NEXT f
 470 DATA 0,0,0,15,240,128,128,1
92,0,0,1,14,240,128,128,192
 480 DATA 0,0,3,28,224,128,128,1
92,0,1,6,56,192,128,128,192
 490 DATA 2,12,16,32,192,128,128
,192,4,8,16,32,192,128,128,192
 500 DATA 18,74,42,169,87,171,95
,47,68,85,91,162,91,87,255,255,2
14,228,169,218,229,254,232,240
 600 BEEP .5,0: BEEP .5,5: BEEP
.7,11: INK 0: PAPER 5: BRIGHT 1:
 BORDER 2: CLS
 605 LET victoria=0: LET derrota
=0: LET puntos=0
 610 PRINT AT 5,5; FLASH 1; INK
2; PAPER 6;"  T A N Q U E S  "
 620 PRINT AT 10,3;"?Que tipo de
 guerra desea?"''TAB 6;"Diurna:1
  Nocturna:2"
 630 INPUT z
 640 IF z>2 OR z<1 OR z<>INT z T
HEN BEEP .5,-20: GO TO 630
 650 CLS : PRINT AT 2,3;"Mueve l
a pieza de artilleria    pulsand
o:"'TAB 15;" o : arriba"''TAB 15
;"j : abajo"''TAB 15;"f : fuego"
 660 PRINT ''"La puntuacion esta
ra de acuerdo a la distancia del
 tanque"''" ATENCION:No se puede
 disparar  continuamente"''" De
 noche guiate por la luz del  di
sparo"
 670 PRINT " Suerte....  Pulsa u
na tecla": PAUSE 0
1700 GO TO (1800 AND z=1)+(1850
AND z=2)
1800 BRIGHT 1: INK 0: PAPER 4: B
ORDER 3: CLS
1810 FOR r=0 TO 10: PRINT PAPER
5; INK 1;AT r,0;"
                 ": NEXT r
1830 PLOT 0,100: DRAW 10,-5: DRA
W 30,20: DRAW 20,-10: DRAW 40,0,
PI/5: DRAW 20,20: DRAW 20,-15: D
RAW 30,-5: DRAW 10,10: DRAW 10,-
15: DRAW 35,40: DRAW 15,-20: DRA
W 5,7: DRAW 10,-25
1840 CIRCLE INK 6;40,155,17: CIR
CLE INK 6;40,155,15
1845 GO TO 1910
1850 BRIGHT 1: PAPER 0: INK 0: B
ORDER 4: CLS
1855 INK 7
1860 PLOT 0,100: DRAW 10,-5: DRA
W 30,20: DRAW 20,-10: DRAW 40,0,
PI/5: DRAW 20,20: DRAW 20,-15: D
RAW 30,-5: DRAW 10,10: DRAW 10,-
15: DRAW 35,40: DRAW 15,-20: DRA
W 5,7: DRAW 10,-25
1870 PLOT 0,90: DRAW 255,0
1880 PLOT 200,170: DRAW 0,-30,PI
*1.2: DRAW 0,30,-PI/1.2
1890 INK 0: PAPER 0
1900 REM COMIENZO DEL JUEGO
1910 LET di=0
1920 IF z=2 THEN PRINT INK 0; PA
PER 0;AT 20,4;"
       "
1930 PRINT INK 2; PAPER 6;AT 21,
0;"Victoria:";victoria;AT 21,11;
"Derrota:";derrota;AT 21,21;"Pun
tos";puntos
2000 LET d=156+INT (RND*6)
2010 PRINT INK 2*z;AT 19,0;"[redacted]";
AT 20,0; INK (-2+z*2);"[redacted]   "
2020 FOR p=29 TO 4 STEP -1
2030 FOR t=153 TO 144 STEP -3
2040 LET di=di-1
2050 PRINT INK (-8+8*z);AT 20,p;
CHR$ t;CHR$ (t+1);CHR$ (t+2)
2060 BEEP .0013,50
2070 REM POSICION DISPARO
2080 IF INKEY$="o" AND d<=160.8
THEN LET d=d+.2
2090 IF INKEY$="j" AND d>156.2 T
HEN LET d=d-.2
2100 PRINT INK 2*z;AT 19,1;CHR$
d
2120 IF INKEY$<>"f" OR di>0 THEN
 GO TO 2270
2200 REM DISPARO
2210 BEEP .00127,50: LET di=9
2215 IF d<160.8 THEN LET d=d+.2
2220 PLOT 16,22
2230 DRAW INK 2; OVER 1;(d-155)*
39,-14,-PI/ABS (d-163)
2240 PLOT OVER 1;16,22
2250 DRAW INK 2; OVER 1;(d-155)*
39,-14,-PI/ABS (d-163)
2260 IF POINT ((d-155)*39+16,8)=
1 THEN LET victoria=victoria+1:
LET puntos=puntos+100*p: PRINT I
NK 2; PAPER 6; FLASH 1;AT 20,p;"
";AT 18,p;100*p: FOR b=40 TO
50 STEP .3: BEEP .00127,b: OUT 2
54,RND*255: BEEP .002,b*SGN (b-1
8): OUT 254,RND*255: NEXT b: PRI
NT AT 20,p;"   ";AT 18,p;"   ":
 GO TO (1900 AND victoria<=9)+(6
00 AND victoria=10)
2270 NEXT t
2280 NEXT p
2290 LET derrota=derrota+1: IF d
errota=5 THEN GO TO 3000
2300 PLOT 31,13: DRAW INK 2;-21,
3: BEEP .5,-30: PLOT 31,13: DRAW
 OVER 1;-21,3: PRINT AT 19,0; FL
ASH 1; INK 2; PAPER 6;"  ";AT 20
,0;"   ": FOR x=30 TO 31 STEP .0
2: BEEP .0013,x: OUT 254,RND*255
: NEXT x: GO TO 1900
3000 INK 7: PAPER 0: BORDER 1: C
LS
3010 PRINT AT 10,3;" Perdiste la
 guerra.RINDETE"
3020 PRINT '''"Pulsa una tecla":
 PAUSE 0: GO TO 600
```

LAS TORRES DE HANOI

Javier EIRIZ AGUILERA

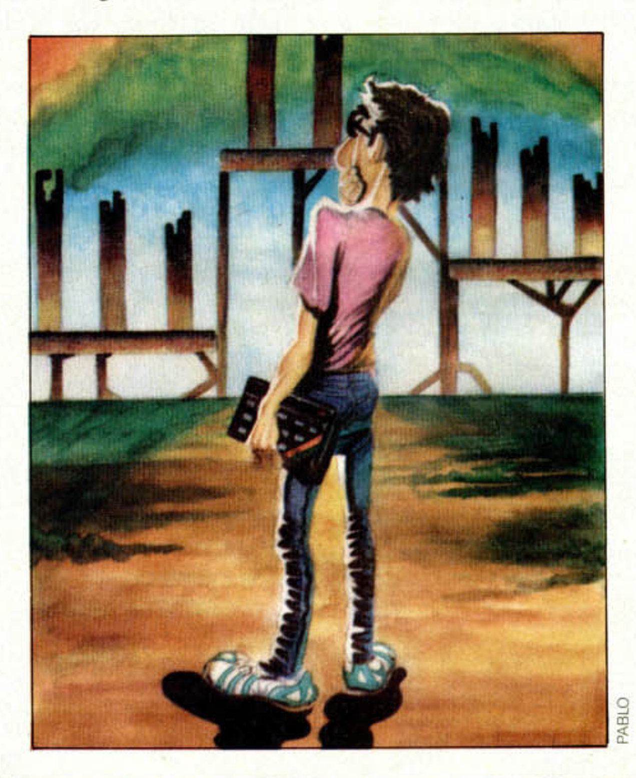
Spectrum 16 K

Premiado con 15.000 ptas.

No se trata de un lugar exótico, ni de viajes excitantes que nuestra imaginación pueda engrandecer. Es, simplemente, un juego en el que tendremos que formar las torres que hemos bautizado con ese sugestivo nombre.

COmo su título indica, nuestra labor en esta ocasión, es la de formar tres torres o bloques hasta conseguir igualar su altura. Para ello, deberemos transportar bloques de una torre a otra, consecutivamente, hasta lograrlo mediante los nueve niveles de dificultad que encierra el programa.

La única condición que se impone en nuestra labor, consiste en la imposibilidad de trasladar un piso más largo sobre otro más corto.


PABLO

```
   1 REM © Javier Eiriz Aguilera
   2 REM      para  MICRO HOBBY
  10 PAPER 1: INK 5: BORDER 1: C
LS
  20 PRINT AT 2,8;"* Torres de H
anoi *"
  30 INPUT "dificultad (2-9)?";d
  40 IF d<2 OR d>9 THEN GO TO 30
  45 CLS
  50 PRINT AT 4,4;"Numero minimo
 de jugadas ";2↑(d-1)
  55 DIM h(3): DIM c(3,d)
  56 FOR f=1 TO d: LET c(1,f)=d+
1-f: NEXT f
  57 LET h(1)=d: LET h(2)=0: LET
 h(3)=0
  58 LET j=0
  60 LET a$="[redacted]"
 100 PRINT INK 3;AT 18,2;"[redacted]";TAB 12;"[redacted]";TAB 22;"[redacted]"
 110 PRINT AT 19,2;"1";TAB 12;"2
";TAB 22;"3"
 115 GO SUB 320
 200 INPUT "desde ";o,"hasta ";i
 205 IF o>3 OR o<1 OR i>3 OR i<1
 THEN GO TO 200
 210 IF h(o)=0 THEN GO TO 200
 215 IF h(i)=0 THEN GO TO 230
 220 IF c(o,h(o))>c(i,h(i)) THEN
 GO TO 200
 230 LET j=j+1: BEEP .2,23
 256 LET c(i,h(i)+1)=c(o,h(o))
 257 LET c(o,h(o))=0
 260 LET h(o)=h(o)-1
 270 LET h(i)=h(i)+1
 275 PRINT AT 17-h(o),1+1*(o=1)+
11*(o=2)+21*(o=3);"         "
 280 GO SUB 320
 300 IF h(2)=d OR h(3)=d THEN GO
 TO 500
 310 GO TO 200
 320 LET w=-1
 330 FOR f=1 TO d: LET w=w+1: PR
INT AT 17-w,2;a$( TO c(1,f));AT
17-w,12;a$( TO c(2,f));AT 17-w,2
2;a$( TO c(3,f)): NEXT f
 340 PRINT AT 21,10;"Jugadas ";j
 350 RETURN
 500 PRINT AT 21,0;"Fenomeno lo
has logrado en ";j
 510 FOR f=1 TO 12: FOR n=0 TO 8
: BEEP .1,n: BEEP .1,f: NEXT n:
NEXT f
 520 PAUSE 0: CLS : PRINT "si qu
ieres jugar pulsa una letrarapid
amente o el programa hara  PLUF!
  en 10 segundos"
 530 FOR f=0 TO 1000: IF INKEY$<
>"" THEN GO TO 1
 540 NEXT f: NEW
```



PONZANO Nº 25
TFN. (91) 441 16 51
28003 MADRID

AGRADECE A







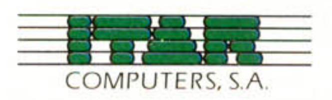
SOFTWARE CENTER



M. C. Aguilar

R.T.V. Gutiérrez

Brasilia Hi-Fi



TEK MICROSISTEMAS





MICROELECTRONICA



Electrónica





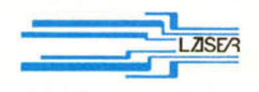








GaherShop





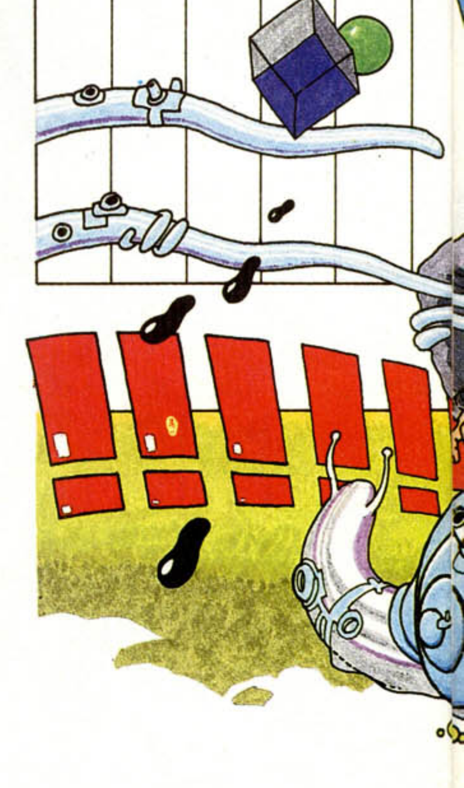


Y a las innumerables personas que comenzando por orden alfabético en D. Salvador Abad y terminando en D. Ramón Zurdo Sánchez, han depositado su confianza en nosotros y nuestros programas desde toda España durante 1984.
Deseando seguir contando con su apoyo en este nuevo año, a todos,

GRACIAS

CARGA Y ALMACENAMIENTO DE PROGRAMAS CON VELOCIDAD VARIABLE (II)

Paco MARTIN y José María DIAZ



Esperamos que el artículo anterior haya cumplido la misión de aclarar al lector el método empleado para analizar la sintaxis de los nuevos comandos que se nos ocurra incluir en el Basic del Spectrum.

Partiendo de esta premisa, presentamos en este número la rutina de sintaxis completa, es decir, con unas modificaciones necesarias para mayor comodidad de uso, pero que de incluirlas en la anterior exposición, pensamos que hubieran dificultado en extremo la comprensión de la lógica del programa.

Asimismo, hemos incluido también parte de la rutina de velocidad variable para que se pueda ir examinando con el detalle y la tranquilidad que requiere, dada la cierta complejidad de este proceso, en el que se mezclan, siguiendo una lógica precisa, partes de la rutina ROM del ordenador modificadas de acuerdo a nuestros propósitos, con trozos de código que ha sido necesario «fabricar», bastante repletos de trucos y artimañas de programación, que esperamos sean útiles a los programadores ávidos de lenguaje máquina.

Hemos tratado de explicar, con la mayor precisión y brevedad posibles, lo que está sucediendo en el microprocesador mientras ejecuta, paso a paso la rutina, esperando, no obstante, que el lector consiga una visión de conjunto, a vista de pájaro, del procedimiento como un todo.

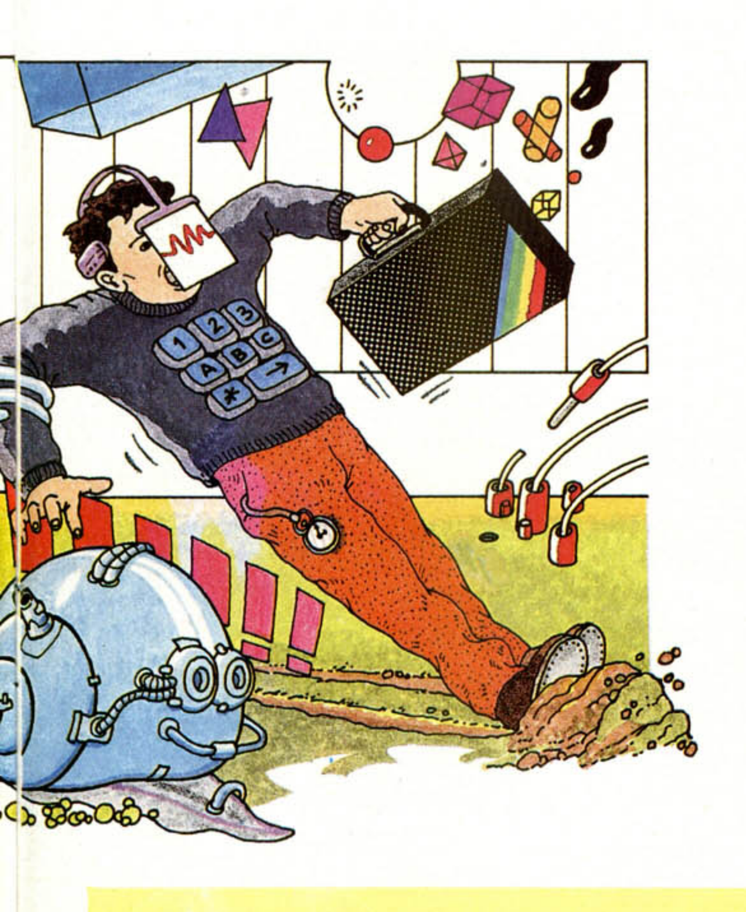
Antes de terminar, dos aclaraciones:

a) para mayor comodidad de uso, después del "REM" deben colocarse dos puntos; así obtendremos fácilmente los "TOKENS" de SAVE, LOAD, VERIFY o MERGE.

b) la rutina está pensada para que a velocidad normal (1.500 baudios) sea plenamente compatible con el sistema operativo del ordenador; sin embargo, quisiéramos matizar de nuevo que a altas velocidades de transferencia de información, la precisión con la que este proceso ocurre, depende de las características físicas del aparato de cassette utilizado.

```
        ORG   6ØØØØ
        LD    HL,(23645)        TOMA LA DIRECCION CH-ADD PARA SINTAXIS INICIAL.
        INC   HL
        LD    A,(HL)
        CP    234 ; 'REM'       COMPRUEBA LA PRESENCIA DE "REM"
        JR    NZ,ERRORA         Y ":"
        INC   HL
        LD    A,(HL)            SI NO EXISTEN, SALTA A ERROR
        CP    ":"
        JR    NZ,ERRORA
        INC   HL
        LD    A,(HL)            EXAMINA LOS SIGUIENTES
        CP    248 ; 'SAVE'      CARACTERES DE LA LINEA DE
        JR    Z,TSAVE           COMANDO PARA BIFURCAR A
        CP    239 ; 'LOAD'      LA SUBRUTINA CORRESPONDIENTE.
        JP    Z,TLOAD
        CP    214 ; 'VERIFY'
        JP    Z,TVERIF
        CP    213 ; 'MERGE'
        JR    Z,TMERGE
        JR    ERRORA            SI NINGUNO EXISTE, SALTA A ERROR.
ERROR6  RST   8                 LLAMADAS A LA SUBRUTINA ROM DE MENSAJES DE ERROR
        DEFB  5 ; 'NUMBER' TOO BIG'
ERRORA  RST   8
        DEFB  9 ; 'INVALID ARGUMENT'
ERRORB  RST   8
        DEFB  10 ; 'INTEGER OUT OF RANGE'
ERRORC  RST   8
        DEFB  11 ; 'NONSENSE IN BASIC'
ERRORF  RST   8
        DEFB  14 ; 'INVALID ARGUMENT'
ERRORR  RST   8
        DEFB  26 ; 'TAPE LOADING ERROR'
TSAVE   CALL  BAUDIO            TOMA EL NUMERO INDICADOR DE LA VELOCIDAD DE GRABA-
```

```
        CALL  NAME              CION. NOMBRE Y SI SU LONGITUD ES
        JR    C,ERRORF          CERO O MAYOR QUE 1Ø , SALTA A ERROR.
        CALL  SINTAX            COGE EL RESTO DE LOS DATOS.
        PUSH  AF                PRESERVA EL INDICADOR DE "TIPO".
        LD    A,#FD             ABRE CANAL Ø
        CALL  #16Ø1
        XOR   A
        LD    DE,#9A1           MENSAJE "START TAPE..."
        CALL  #CØA
        SET   5,(IY+2)          FLAG PARA BORRAR MENSAJE.
        CALL  #15D4             ESPERAR PULSACION TECLA.
        LD    IX,CABEC          PUNTERO PARA EL BUFFER DE CABECERA.
        LD    DE,17             LONGITUD DEL BUFFER.
        XOR   A                 INDICA "ES UNA CABECERA".
        CALL  SAVE              SALVA LA CABECERA.
        LD    B,#32
        HALT                    SE PRODUCE UN RETARDO ANTES DE SALVAR EL SEGUNDO
        DJNZ  PSAV              BLOQUE.
        POP   AF                RECUPERA "TIPO"
        LD    DE,(LONT)         LONGITUD DEL BLOQUE A SALVAR.
        LD    A,#FF             SEÑAL "BLOQUE DE DATOS".
        LD    IX,(23635)        SI ES UN PROGRAMA, TOMA DIR. DE COMIENZO.
        JR    C,FSAV
        LD    IX,(DIRT)         SI NO, LA DIRECCION ESPECIFICADA EN EL COMANDO.
FSAV    CALL  SAVE
        RET
TMERGE  CALL  BAUDIO
        CALL  NAME
        CALL  C,CPNAME          COMPRUEBA QUE EL FINAL ES ' " '
        INC   HL
        LD    A,(HL)            AL FINAL DEBE HABER UN "ENTER".
        CP    13
        JR    NZ,ERRORC
        CALL  RELOAD            CARGA LA CABECERA.
```



```
         LD     BC,(LONBUF)      LONGITUD TOTAL DEL PROGRAMA.
         PUSH   BC
         INC    BC
         RST    48               CREA EN EL AREA DE TRABAJO "BC+1" ESPACIOS.
         LD     (HL),#8Ø         PONE UN INDICADOR DE FIN EN LA LOCALIZACION EXTRA.
         EX     DE,HL
         POP    DE
         PUSH   HL
         PUSH   HL
         POP    IX
         LD     A,#FF
         CALL   LMBYT            CARGA EL BLOQUE DEL PROGRAMA.
         JP     2253             EFECTUA LA OPERACION DE "MERGE".
TVERIF   CALL   LDPREV           INICIALIZA VALORES.
         RES    Ø,(IY+37)        SEÑAL "VERIFICANDO".
         JR     NC,LDBYTE        SALTA SI NO ES UN PROGRAMA BASIC.
         JR     CPROG            VERIFICA EL PROGRAMA BASIC.
LDPREV   CALL   BAUDIO           TOMA LOS VALORES DEL COMANDO.
         CALL   NAME             COMPRUEBA SINTAXIS Y CARGA LA CABECERA.
         CALL   C,CPNAME
         CALL   NSINT
         PUSH   AF               PRESERVA EL INDICADOR DE TIPO.
         CALL   RELOAD
         POP    AF               LO RECUPERA.
         RET
TLOAD    CALL   LDPREV
         SET    Ø,(IY+37)        SEÑAL "CARGA EL BLOQUE".
         JR     NC,LDBYTE        SALTA SI NO ES UN PROGRAMA BASIC.
         LD     DE,(LONBUF)      TOMA LONGITUD TOTAL DEL BLOQUE A CARGAR.
         LD     HL,(23635)
         ADD    HL,DE            MIRA SI EL PROGRAMA A CARGAR ES MENOR QUE EL PRO-
         EX     DE,HL            GRAMA RESIDENTE.
         LD     HL,(23641)
         SCF
         SBC    HL,DE
         JR     C,ADDS           SI ES MAYOR O IGUAL, SALTA
         JR     Z,ADDS           PARA CREAR ESPACIO.
         LD     B,H              RECLAMA LOS BYTES SOBRANTES.
         LD     C,L
         EX     DE,HL
         CALL   #19E8
         JR     PPL
ADDS     ADD    HL,DE            CREA ESPACIO SUFICIENTE EN MEMORIA PARA EL NUEVO
         EX     DE,HL            PROGRAMA.
         AND    A
         SBC    HL,DE
```

```
         LD     B,H
         LD     C,L
         EX     DE,HL
         CALL   #1655
PPL      LD     HL,(23635)       ASIGNA LA DIRECCION CORRECTA DE LAS VARIABLES DE
         LD     BC,(INFBUF)      PROGRAMA.
         ADD    HL,BC
         LD     (23627),HL
         LD     HL,(DIBUF)       MIRA SI EL PROGRAMA BASIC SE AUTOEJECUTA Y EN QUE LI-
         LD     A,H              NEA.
         AND    #CØ
         JR     NZ,CPROG
         LD     (23618),HL       SI ES ASI, HACE UNA COPIA.
         LD     (IY+1Ø),Ø        SEÑALA PRIMERA INSTRUCCION DE LA LINEA.
CPROG    LD     DE,(LONBUF)
         LD     IX,(23635)
CARGA    LD     A,#FF
         BIT    Ø,(IY+37)        EXAMINA "LOAD O VERIFY".
         JR     Z,VBYT
LMBYT    SCF
VBYT     CALL   LOAD
         RET    C
         JP     ERRORR
LDBYTE   AND    A                SE EFECTUA UN CHEQUEO DEL REGISTRO "A" PARA DETER-
         JR     NZ,INDEF         MINAR EL TIPO DE CARGA
         LD     IX,16384         A=Ø INDICA "LOAD SCREEN$".
         LD     DE,(LONT)        A=1 INDICA "LOAD CODE".
         LD     HL,(LONBUF)      A=2 INDICA "LOAD CODE XXXX".
         AND    A                A=3 INDICA "LOAD CODE XXXX, XXXX".
         SBC    HL,DE
         JP
         JR     CARGA
INDEF    CP     2
         JR     NC,(DIDEF)
         LD     IX,(DIBUF)
DEFLO    LD     DE,(LONBUF)
         JR     CARGA
DIDEF    LD     IX,(DIRT)
         JR     NZ,TODEF
         JR     DEFLO
TODEF    LD     DE,(LONT)
         JR     CARGA
RELOAD   LD     IX,BUFCAB        DIRECCION DE CARGA DE LA CABECERA.
         LD     DE,17
         XOR    A
         SCF
         CALL   LOAD
         JR     NC,RELOAD        SI ERROR, INSISTE.
         RES    Ø,(IY+82)        ABRE CANAL SUPERIOR DE PANTALLA.
         LD     82               PREVEE SCROLL DE TRES LINEAS.
         LD     HL,CABEC         DIRECCION PUNTERO CABECERA ESPECIFICADA.
         LD     C,128            SEÑAL "DIFERENTE TIPO".
         LD     A,(BUFCAB)
         CP     (HL)             COMPARA AMBAS CABECERAS.
         JR     NZ,LDTIP         SALTA SI NO SON IGUALES.
         LD     C,246            SEÑAL "1Ø CARACTERES IGUALES".
LDTIP    CP     4                SI EL TIPO DE BLOQUE ES MAYOR
         JR     NC,RELOAD        QUE 3,CARGA NUEVA CABECERA.
         LD     DE,#9CØ
         PUSH   BC               IMPRIME EL MENSAJE ASOCIADO A CADA TIPO DE BLOQUE
         CALL   #CØA             Y EL NOMBRE DEL BLOQUE ENTRANTE.
         POP    BC
         LD     DE,BUFNAM
         LD     HL,NOMBRE
         LD     B,1Ø
         LD     A,(HL)
         INC    A
         JR     NZ,LDNAME
         LD     A,C
         ADD    A,B
         LD     C,A
LDNAME   LD     A,(DE)
         CP     (HL)
         INC    HL
         INC    DE
         JR     NZ,LDPR
         INC    C
LDPR     RST    16
         DJNZ   LDNAME
         BIT    7,C              EXAMINA SI EL TIPO Y EL NOMBRE
         JR     NZ,RELOAD        ENTRANTES COINCIDEN.
         LD     A,13
         RST    16
         RET
```

LOS MUCHACHOS DEL «FRED»: UN TRABAJO COMPARTIDO

Jesús ALONSO GALLO

La habitación donde programan estos muchachos, parece una leonera. Todo está revuelto y aparentemente desorganizado. Sin embargo, Charly me comenta: «Bueno, al saber que venías esta tarde, lo hemos recogido todo un poco. Normalmente está bastante peor.»

El equipo, bien sencillo: dos spectrum conectados entre sí, una impresora y una unidad de Microdrive. Al escribir estas líneas, caigo en la cuenta de una realidad fácilmente constatable: los mejores equipos de programación de España trabajan con unos medios modestísimos, y rápidamente me surge la pregunta: ¿Qué programas conseguirían hacer todos estos programadores españoles si contasen con los medios que poseen las empresas punteras de Inglaterra? La pregunta queda en el aire, sois vosotros los lectores quienes tenéis que buscar la respuesta.

Se me ocurrió que lo mejor sería una presentación personal de cada uno.

A partir de este momento lo que sucedió en aquella habitación fue realmente divertido: yo abandoné mi papel de periodista, ellos olvidaron su situación de entrevistados, y sin hacer preguntas tocamos muchos temas de interés. Dejándonos llevar por una conversación anárquica, conseguí personalmente un éxito profesional y todos nos reímos a pleno pulmón.

«Yo soy Carlos Granados, conocido como Charly por esta peña e hice el Fred, primero para Spectrum, y el Fred para Amstrad, y ahora estamos haciendo otro programa nuevo para Spectrum.»

«Yo soy Paco Menéndez y, lo mismo que Charly, trabajo desde el principio.»

—Pero ¿trabajáis el programa en conjunto? ¿No distribuís la labor por partes?, pregunto.

—«¿Quieres decir partes definidas?

—Sí, y Charly responde con soltura: «Bueno yo me encargué más de los gráficos de los muñequitos porque se me da mejor dibujar, pero las decisiones se toman entre todos. Todo se hace en grupo, la mayoría suele ser consenso.»

«Yo me llamo Camilo Cela y empecé en el segundo Fred.»

«Yo soy Fernando Rada, comencé desde el principio como ellos.»

—¿Qué edades tenéis?

Sinceramente esperaba una variedad de edades entre ellos porque yo juzgaba a unos mayores que otros. Sin embargo, me responden muertos de risa:

—«Tenemos todos 19 años.»

Parece como si en un principio Charly fuese el más animado a conversar y a él me dirijo:

—¿Cuándo cogiste tu primer ordenador?

Para estos jóvenes programadores, la dedicación a la Informática es un hobby.

—«A los 14 ó 15 años.»

—¿Era el ZX 80 o el ZX 81?

—«No, yo el primer ordenador lo cogí en EE.UU.» (Nuevo corte que despierta mi curiosidad).

«Me fui allí de intercambio para aprender inglés y tal, resulta que donde estaba tenían una máquina, era gigantesca y horrible, no me acercaré más en mi vida a una cosa como esa. —Reíamos todos—, allí hice mi primer programa en



Un equipo unido por una misma pasión.

BASIC, una máquina tragaperras. Después, al volver a España, tuvimos la suerte de ser uno de los primeros institutos que comenzaron a impartir clases de Programación en Basic para ordenadores.»

La conversación se hace cada vez más fluida y, hablando sobre las primeras experiencias con otras máquinas, llego a la conclusión de que las personas que tienen un alto nivel de programación, han pasado en su mayoría por el ZX 80 o ZX 81. La explicación de todo esto es variada, por un lado con el ZX 80 o se programa o se tira la máquina a la basura, no existía software de ningún tipo, había que inventarlo. Otra razón es aducida por Charly:

«Aunque fuese un palazo de Ivestrónica 42.000 Pts. por el ZX 81, cuando llegó la máquina a España, no había nada de menos de 150.000 Pts.»

Su dedicación a la informática está orientada como hobby, ellos estudian carreras universitarias, Físicas unos y otros Ingeniero de Telecomunicaciones. Pablo Menéndez me comenta: «Es que al principio, éstos empezaron antes que yo, lo tomábamos como hobby, yo empecé en el colegio y como se me daba bien, pues les conocí y empecé a meterme en la historia ésta. Ellos hicieron un programilla para la ZX 81, uno tipo PACKMAN con métodos rudimentarios sin ensamblador ni nada, pokeando y tal. Después entramos a trabajar en Indescomp de casualidad; primero entraron a trabajar ellos



La gestión democrática y una perfecta conjunción, son la clave de un trabajo en equipo.

tres con el hermano de Charly y un amigo suyo. Fue una historia porque estaban en el SIMO y vieron una ZX 81 y se pusieron a juguetear con ella, les vieron gente de Indescomp y les dijeron que se pasasen por la tienda, fueron y les propusieron trabajar allí. Comenzaron a trabajar ellos tres en Enero del 83.

—¿En qué régimen trabajabais?

—«Trabajábamos con un sueldo mensual, íbamos allí tres días a la semana y trabajábamos tres horas. Era un horario muy cómodo y flexible.» —Me responde Charly como una bala—.

—¿Qué hacíais y qué os pagaban?

—«Nos pagaban 17.000 Pts. al mes y nos dedicábamos a traducir programas, desprotegerlos. Nuestra labor era programar la entrada de la Spectrum en España, también hicimos cosas para el Dragón que luego salió, como los 50 mejores programas para el Dragón.»

—Hablando de vuestra mecánica de trabajo, cuando estáis programando ¿cómo surgen las ideas?

Alguien salta como un resorte y en tono academicista comenta:

—«Te puedo decir que todo lo que hacemos es empírico 100%. Decimos: Vamos a hacer un programa este año, y ya está.»

Me quedo tan atónito que ellos lo notan y continúan explicándome.

«Primero surge una idea general y, después, surgen ideas parciales que conforman el programa, que si el muñeco debe correr hacia adelante y atrás, que si esto debe hacerse de tal o cual manera, etc. Todas las cuestiones se deciden entre todos.»

Gestión democrática

Les gusta dejar claro el tema de la absoluta gestión democrática que existe dentro del grupo. La conversación se sucede de una forma muy concatenada y yo intento no cortar sus intervenciones.

«Con Fred sucedió que Charly llegó en verano con una idea para un programa. La cosa consistía en un muñequito que subía y bajaba por una cuerda, con esto empezamos a trabajar.»

—¿Por qué el Fred no llevaba protecciones?

—«Realmente fue un problema de tiempo material, entregamos el programa y el tema de las protecciones no se hizo.»

—¿Cuánto tiempo duró el desarrollo del Fred?

—«Desde la idea inicial, 4 meses, pero realmente, en hacerlo 15 días. Charly —dice Paco Menéndez— empezó primero el muñequito, después la subrutina del laberinto y, finalmente, nos volcamos todos en hacer el programa completo.»

La idea del «Fred» surgió de Charly y duró cuatro meses su elaboración.

—Me gustaría preguntaros qué opináis sobre IBM.

—Charly responde de una forma genial que me deja admirado: «Cuando IBM haga algo como el Spectrum, sabremos lo que es». Realmente es una respuesta sólida.

De todo lo comentado me sorprende la idea que ellos dan al grupo, el equipo unido, en perfecta conjunción, decide lo que se hace, el trabajo de todos es sometido a la crítica de todos, nadie se desconecta, todos tienen entre manos alguna subrutina que depurar.

Sobre el QL opinan que parece no estar colmando las expectativas depositadas en él, sin embargo, no creen que sea un mal ordenador. Fernando Rada afirma: «Es demasiado caro para ser una máquina de juegos y demasiado lento para ser una máquina de gestión».

Comentamos el tema de la calidad de nuestros programadores respecto a los ingleses y ellos me dicen que en España los que programan son cuatro gatos que tienen un gran nivel. En Inglaterra programan muchísimas personas y sólo tres o cuatro son buenos, sin embargo, los de mala calidad también ganan dinero.

—¿Qué os parece todo el tema de las nuevas tecnologías?

Aquí cada uno decía una cosa. Para Fernando Rada, algunos temas iban a fracasar, otros decían que pronto habría un gran desarrollo en el campo de aplicaciones de la fibra óptica con fines informáticos. Llegando este punto, la entrevista se disolvía cada vez más rápido y surgía la conversación.

—¿Cuántos habéis vendido?

—«En Inglaterra, probablemente más de 20.000.»

Revelándome secretos inconfesables sobre el nuevo programa en el que trabajan y sobre investigaciones secretas para evitar el pirateo, llega el momento de apagar el mudo testigo de nuestra charla para que yo guarde en mi cofre de periodista los misterios y, vosotros, lectores, la curiosidad. Sólo os adelantaré una cosa: EL NUEVO PROGRAMA EN EL QUE TRABAJAN ES ABSOLUTAMENTE FANTASTICO.

CONSULTORIO

Juegos comercializados

Los juegos que se están vendiendo en el mercado ¿llegaron a salir publicados en la revista?; ¿cuándo aproximadamente?; ¿hay alguna pega para los programas de 16 y 48 K en el de 64 K?

Fernando CARMONA-Madrid

☐ Lamentamos comunicarle que no podemos publicar los listados de los programas comerciales por razones obvias, entre ellas, el respeto a la propiedad privada. No existe pega alguna para que los programas de 16 K se ejecuten en un Spectrum de 64 K.

«EDIT»

Les ruego que me informen cómo se pone «edit» en el ordenador. Como sugerencia, «por qué no ponen un dibujo de cómo es el programa cada vez que publican uno nuevo?

Juan José BLANCO-Valladolid

☐ Edit no es ni un comando ni una función, sino una utilidad del sistema operativo del ordenador que permite editar líneas de programa para corregirlas en la parte inferior de la pantalla. Esto se consigue pulsando a la vez CAPS SHIFT y 1.

Agradecemos su sugerencia y la tendremos en cuenta para próximas publicaciones.

Para 48 K

Ya he leído en el n.º 3, que desde ahora vais a especificar en los programas la cantidad de memoria requerida para su elaboración, pero quiero que me digáis para cuantos Kb son los siguientes programas. Aventura en la Pirámide, Un día en las carreras, Aritmemori, Pipeline y Andrómeda.

Fco. José SEGOVIA-Madrid

☐ Todos los programas que usted nos indica son para Spectrum de 48 K.

Programa de fútbol

Soy un adicto a los programas deportivos, pero no encuentro ningún programa de Fútbol para Spectrum, ¿hay programas de este tipo para el Spectrum? Si los hay, ¿me podrían decir dónde los puedo adquirir?.

Federico SANCHEZ-Málaga

☐ Existe un programa llamado WORLD CUP que puede usted encontrar en KEY INFORMATICA al precio de 1.000 pesetas. La dirección de la tienda la encontrará usted en cualquiera de nuestras revistas.

Programas para los Spectrum

Me gustaría que me informasen si los programas de lectores y los programas MICROHOBBY, que publican en su revista, se pueden introducir en un Spectrum + sin ninguna modificación en ellos.

Gabriel CASIELLES-Asturias

☐ Efectivamente, se pueden introducir en el Spectrum sin ningún tipo de modificación; recuerde que, como otras veces hemos comentado, ambos ordenadores son idénticos, excepto en el teclado.

Movimiento de pantalla

Quisiera saber cuáles son las direcciones de memoria o rutinas que permiten el movimiento de la pantalla en horizontal (Penetrator), en cuatro direcciones (hormigas) y del modo frontal (Peatchase Banda a cuadros).

J.L.E.G.-Alicante

☐ Algunas de esas rutinas que usted dice se encuentran en la ROM del Spectrum, pero otras de los programas comerciales que usted comenta, son hechas artesanalmente por los autores de dichos programas, por lo cual no podemos publicarlas ni darle información acerca de ellas. Respecto a las existentes en ROM, cualquier libro que tenga la ROM desensamblada del Spectrum podría servirle.

Plantillas para el Spectrum

He visto muchos anuncios que regalan o venden plantillas para el teclado del Spectrum, ¿qué son y para qué sirven dichas plantillas?

Javier FERNANDEZ-Cádiz

☐ La plantilla es, normalmente, una plancha de cartón agujereada de forma tal, que encaja en el teclado del ordenador y contiene indicaciones útiles para programas de aplicaciones o de juegos.

Programa Ensamblador

Quería preguntarles si al suscribirme recibiré una revista cada semana o 4 al mes.

¿Cómo puede introducir en mi Spectrum los programas que vienen en Asambler o Código máquina?

Jesús TEJERO-Salamanca

☐ Nuestros suscriptores reciben un número de la revista cada semana.

Para introducir en su ordenador los programas que vienen en ensamblador o código máquina, necesita Vd. comprarse un programa ensamblador; encontrará usted multitud de ellos de distintos tipos y prestaciones en las tiendas de informática.

Grabar programas

¿Se puede grabar un programa en un radio-cassette igual que en una grabadora?

José M.ª LORENTE-Valencia

☐ La respuesta es sí. Los programas pueden salvarse y cargarse en cinta de cassete convencional y el aparato no requiere condiciones especiales.

La necesidad de Interfaces

Me gustaría saber si se puede colocar un mando para los juegos sin tener que comprarme un «Interface». Todo esto ¿cuánto me costaría?

Pablo GONZALEZ-Madrid

☐ Mucho nos tememos que no va a poder hacerlo. Tendrá que comprarse un Interface. Toda la información que nos solicita acerca de precios y modelos la tiene en el número 8 de nuestra revista.

Transcripción de programas

Tengo un Atari 800 xl, y mi pregunta es la siguiente: ¿existe algún método para poder transcribir los programas de Microhobby?

Luis CIGES-Valencia

☐ Lo que usted nos plantea reviste una gran dificultad a la hora de dar una respuesta categórica; la razón estriba en que las dos máquinas son radicalmente distintas. Conociendo suficientemente el Basic del Spectrum, creemos que si podría «traducir» los programas de un ordenador a otro a mano, por supuesto; ahora bien, el asunto se complica en grado sumo en el caso de aplicaciones que se dirijan directamente al Hardware del Spectrum (código máquina, direcciones de pantalla, etc.). Sin pretender desanimarle, podemos decir con justicia que se trata de una tarea de «chinos».

«Editor de caracteres»

Me gustaría hacerles algunas preguntas.

—¿Los listados que publi-

can para ordenador de 16K pueden funcionar en uno de 48?

—El programa que publicaron con el nombre de «Editor de caracteres» no comprendo cómo funciona, ¿qué es lo que hace? Me interesa por los juegos que publican.

Alberto DELGADO-León

□ Efectivamente, los listados para un Spectrum de 16 K pueden funcionar en un 48 K sin ningún proglema.

Para dar satisfacción a la segunda pregunta, necesitaríamos saber con qué parte concreta del programa tiene dificultades de manejo; no obstante, le remitimos a las instrucciones incluidas dentro del programa y en el comentario de presentación de la revista; consideramos que una lectura minuciosa de las mismas resolvería su problema conjuntamente con la utilización del Editor.

16 ó 48 K

Tengo algunos problemillas y espero, si es posible me los solucionéis.

—No consigo encontrar el n.º 1 de MICROHOBBY por ningún lado, ni siquiera en la distribuidora, y como no quisiera que me faltara ningún n.º, os pido si es posible, que me lo enviéis contra reembolso, naturalmente. Yo pagaría los gastos de correo. Si no fuera posible por este sistema, por favor decidme cómo puedo conseguirlo.

—Estoy pensando en comprarme un ZX Spectrum, pero ¿cuál? el de 16 K o el de 48 K ¿qué diferencias existen entre ambos, y que diferencia existe, o que problemas surgirían, si al de 16 K le ponemos la ampliación a 48 K?

—Me ha comentado un amigo, que al conectar el ordenador al televisor la capacidad de memoria disminuye, ¿es esto cierto?, y si es así ¿cuánto disminuye?

—¿Existe en el mercado alguna ampliación de memoria para el ZX 48 K?

Javier MARTIN-Canarias

□ 1. El número uno de Microhobby ya se te ha enviado.

2. La diferencia estriba en la distinta capacidad de memoria RAM útil para el usuario, mucho mayor en el de 48 K.

3. La memoria no disminuye al conectar el ordenador al televisor.

4. Que nosotros sepamos, no existe esa ampliación por el momento.

Derechos de suscripción

Deseo que me resuelvan algunas dudas.

—Si yo envío un programa hecho por mí, ¿recibo 15.000 pts. si se publica?, o lo mandan siempre que reciben un programa.

—Los programas que vendéis copiados de la revista, ¿son todos los programas o algunos?

—¿Tenemos algún derecho los suscriptores de esta revista en cuestión de premios o en algún regalo?

—¿Podríais publicar en la siguiente revista el programa escrito de Piloto de línea (simulador de vuelo)?

Fco. Miguel COLLADO-Sevilla

□ Responderemos a sus preguntas por orden:

1. Recibirá 15.000 pesetas SOLO si su programa se publica.

2. Si se refiere a la cinta que contiene los programas del mes, van incluidos todos los que se publican en la revista.

3. Periódicamente realizamos sorteos entre nuestros suscriptores y, además, éstos se benefician de una reducción en el precio de la revista.

4. No podemos publicar programas comerciales.

Cuestión de Kbites

Me gustaría saber de cuántos Kbites de memoria son los programas «Andrómeda» (del n.º 2) y «Aventura en la pirámide» (del n.º 3) para no hacerlos en vano, ya que poseo un Spectrum 16 K.

Agustín FERNANDEZ-Madrid

□ Los programas que usted nos indica son ambos para 48 k.

Spectrum + o Commodore 64

Estoy indeciso y me agradaría que me aconsejáseis sobre si debo elegir un Spectrum + o un Commodore 64, y qué ventajas o defectos puede tener dicha elección.

Me agradaría saber cuáles son las «medidas» exactas del Spectrum +.

Fco. Javier GARCIA-Hospitalet

□ Nos resulta muy difícil aconsejarle entre los dos ordenadores que usted nos propone, ya que desconocemos cuáles son sus necesidades y sus propósitos.

De cualquier forma, ambas máquinas son totalmente distintas en lo fundamental, es decir, incorporan microprocesadores distintos, los dialectos de basic son radicalmente diferentes, etc.

Sin embargo, ambos tienen aproximadamente, la misma capacidad de memoria RAM utilizable para programas Basic, aunque el Spectrum tiene a su disposición mucha mayor cantidad de programas de todo tipo, especialmene juegos.

Problema de dinero

Me gustaría saber qué ordenador recomiendan para un chico de 14 años con un padre que se quiere gastar poco dinero.

Claudio HUERTA-Barcelona

□ Ya que se ha dirigido Vd. a nuestra revista, es de suponer que se refiere a la gama de Sinclair. Los dos ordenadores más baratos de la misma son el ZX 81 y el Spectrum de 16 K. Como este último ha bajado recientemente de precio, nos parece que es el mas indicado en su caso, con lo que siempre tiene abierta la posibilidad de ampliarle la memoria en un futuro. El ZX 81 sigue siendo una máquina maravillosa, especialmente si consideramos su precio, aunque evidentemente, no puede competir con el Spectrum.

WAFADRIVE

Tento un Spectrum 48 K y me gustaría adquirir un WAFADRIVE. Lo que quiero saber es si este aparato trabaja con cartuchos como los del MICRODRIVE o con Diskettes FLOPPY, etc, y la cantidad de memoria que pueden tener y el tiempo de acceso. También me interesaría saber dónde lo puedo adquirir y el precio aproximado.

Me fio de pedir consejo y les digo: ¿Qué es mejor y más rentable: el MICRODRIVE, el WAFADRIVE, o la nueva Unidad de Disco de INVESTRONICA?

Luis ILLANAS-Torrejón de Ardoz

□ Desde su reciente presentación en el SIMO, no ha sido plenamente comercializado. En líneas generales, el WAFADRIVE utiliza unos cartuchos especiales de doble tamaño que los del Spectrum con una capacidad que oscila entre 16 K y 128 K máximo, según modelos. El tiempo de acceso es sensiblemente mayor que el del Microdrive y su precio aproximado ronda las 48.000 ptas.

En cuanto a su pregunta sobre la rentabilidad de estos aparatos, debe tenerse presente que a mayores prestaciones mayor precio, por lo que corresponde a Ud. tomar la decisión final.

DE OCASION

• VENDO ZX Spectrum 48K con 10 juegos, Harrier Attack, Chequered Flag, pin Ball, Fighter Pilot, Space Raiders, Pyramid, Galaxians, Sabotage, Make a Chip, Androide 2. Por 34.000 pesetas, comprado, en el verano. NUEVO. Nicolás Diaz Lluna, calle del Pilar n.º 40, 2.º C. Santa Cruz de Tenerife. Tel.: 27 23 63. Código Postal 38002.

• DESEARIA intercambiar progamas de 16K, preferentemente en la provincia de Valencia. Llamar al teléfono 280 01 51, y preguntar por José. Llamen preferentemente de 7 PM a 9 PM.

• VENDO ZX Interface 2 a estrenar, por 5.000 ptas. Interesados escribir a: Arturo Cano López. Palermo n.º 44, 2.º 28043 Madrid.

• INTERCAMBIO toda clase de programas para el ZX Spectrum 48/16K. Interesados preguntar por Antonio al teléfono (96) 351 82 76, a partir de las 20,00.

• ME GUSTARIA tomar contacto con usuarios del ZX (48K) Spectrum para intercambio de programas y formar un club. Dirigirse a Joaquín Forner. Avenida España 106. San Jaime de Emveija. Tarragona (Sin teléfono).

• DESEO intercambiar programas para el ZX Spectrum 16/48K. Poseo más de 250 programas en continuo aumento, todos los superventas, ingleses y españoles. Para contactar conmigo llamar al teléfono (91) 889 08 64, desde las 9 de la noche o bien mandar una carta a Alberto Sánchez Hernández. C/Cánovas del Castillo 5, 1.º J. Alcalá de Henares (Madrid).

• VENDO Microdrive + accesorios + garantía, 10.000 pesetas; Interface + accesorios + garantía, 10.000 pesetas; (16K) ZX Spectrum + accesorios + libro, 20.000 pesetas; todo incluido, 35.000 pesetas. Preguntar por Juan, tefno.: (93) 230 48 44, de 2 a 3,30 de la tarde.

• ME GUSTARIA fundar un club de amigos del Sepctrum. Programas, boletín, ofertas, colaboraciones, etc. Escribe y te informaré. Club ZX AMIGOS. Apdo. Correos 28. Villafranca Bierzo (León).

• INTERCAMBIO Programas Spectrum 16/48. Interesados escribir a: José Antonio Suárez Márquez. C/Vázquez Varela 19, 3.º B. Vigo (Pontevedra).

• CAMBIO un órgano-calculadora CASIO VL-TONE con instrucciones para su manejo, por ordenador ZX 81 con todos los accesorios; conectores para la televisión, fuente de alimentación y libro de instrucciones. Mis señas son estas: Rafael González Cárdenas. C/Entrearroyos n.º 6, 3 izquierda. C.P.: 28030. Telf.: 430 29 67. Moratalaz-Madrid-30. (Por favor, las personas que escriban o llamen preferentemente que sean de Madrid), (llamar de 2 a 3 de la tarde).

• VENDO amplificador de sonido (X10) con función RESSET e indicadores luminosos de ON-OFF y LOAD-SAVE (+ conmutador de función) por 2.500 pts. Interface + Joystick por 3.500 pts. Cassette SANYO Reportes con cuentavueltas por 8.000 pts. Tlf.: (91) 246 39 30, Fernando de 3 a 4,30.

• INTERCAMBIO programas para Spectrum 16K/48K: Manic Miner, Jet pac, simulador de vuelo, Bandera a cuadros, Psst, etc. Interesados escribir a: Julián Cervantes. C/Duque de Huete n.º 32. Archena (Murcia). Telfs.: (968) 67 04 27 o bien 67 01 67, llamar de 1 a 3,30 del medio día y por la tarde de 6,30 en adelante.

• VENDO ZX Spectrum 48K (Fuente de alimentación, cables, manual Basic, etc.) Todo ello a precio increíble. También cambio programas. Carlos. Tlf.: (91) 705 98 29. Madrid.

• CLUB Espectros, cambiamos programas (de todas clases) 16K y 48K. Ramón J. Requejo Camiña. Vimbieiro s/n. Sangenjo, Pontevedra. Tel.: (986) 72 35 78.

• VENDO Spectrum 16K con ampliación de 32K externa, cables, transformador y manuales junto con Joystick e Interface de tipo Kemspton, por 40.000 pesetas. Cinta de regalo con 30 juegos. Además, vendo Interface 1, Microdrive con dos cartuchos de regalo con juegos. Esto último con garantía por 37.500. Valentín Arnanz. Avda. Donostiarra, 23 5.º 2. Telf.: (91) 403 20 41. Madrid-28017.

• VENDO ZX Spectrum 16K con sus conexiones, fuente de alimentación y sus manuales en castellano. Regalo varios juegos (Tranzam, Intruders, Meteors, Jumping Jack etc.) Precio a convenir. Pregunten por Jorge en el tlf.: 464 44 21, de 14 a 16 h. Madrid.

• VENDO Spectrum 48K todavía con garantía Investrónica, en pts. 35.000, también cambio mis programas, más de 500, últimas novedades. Escribir a Marco Ortega Monton. C/Alonso Allende 15, 7.º D. Portugalete (Vizcaya).

• INTERCAMBIAMOS programas para Spectrum 16 ó 48K. Buen repertorio de juegos y utilidades. Francisco Javier Pérez Martín. C/Capitán núm. 7, 4.º izd. Aranjuez (Madrid). O al tlf.: (91) 891 17 72.

• CAMBIO juegos de todo tipo (aventuras, inteligentes, etc.) Pregunte por Arturo. Tlf.: (91) 457 99 21. C/Colombia 40, 3.º Madrid-16.

• DESEARIA comprar una ampliación de memoria de 16 a 48K. Intercambio programa de 16K con personas que tenga pocos programas como yo. Telefonear al número 66 31 75, después de las 9 h. y preferible escribir a la dirección: Carlos Maraver Guitar. C/Carlos Marx, Edfi. Cantábrico n.º 5 B.I. Sevilla-41006.

• VENDO Sinclair ZX-81 en perfectas condiciones, con manual de instrucciones en castellano, fuente de alimentación y cables. Todo por sólo 11.000 pesetas. Llamar al teléfono 22 10 13 de Badajoz, preguntar por Carlos.



UN SISTEMA MUSICAL COMPLETO PARA TU ZX SPECTRUM

MELODIAN, EL PROGRAMA MAS VENDIDO DEL MUNDO

El **Melodian** ha sido diseñado para asistir todas las fases de realización musical a los usuarios del **Spectrum** escribiendo, ejecutando, editando, imprimiendo y guardando en cinta magnética. Es instantáneo y fácil de usar, incluso sin tener ningún conocimiento de música. Va provisto de un teclado especial de cartulina que convierte el **Spectrum** en un teclado musical simulado. El sistema utiliza notación real y es una herramienta ideal para la enseñanza y aprendizaje de los principios de la música (notación, ritmos, escalas, intervalos). Es divertido de tocar para los no iniciados y tiene mucho que ofrecer a los músicos profesionales.

REQUERIMIENTOS DE HARDWARE:

ZX Spectrum y opcionalmente una impresora.

CAPACIDAD DE ALMACENAMIENTO:

16 piezas musicales con 254 compases con rutinas especiales SAVE/LOAD para su almacenamiento en cinta.

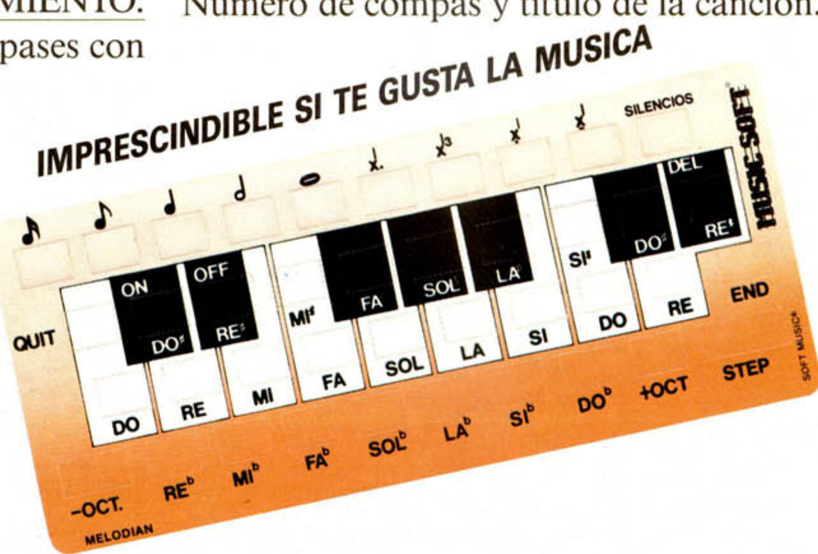
TESITURA: 6 octavas

RITMO Y TIEMPOS:

Tiempo de 23 a 255 (escala de metrónomo. Tiempo real).
Compases de 2/4 a 8/4 y de 2/8 a 15/8.
Control de barras divisorias inteligente.

GRAFICOS:

Todas las figuras, puntillos, tresillos, staccato, tenuto.
Todos los gráficos aparecen simultáneamente con sonido.
2 claves Sol y Fa en 4.ª, armadura y compás.
Notas con sostenidos, bemoles, becuadros, dobles sostenidos.
Indicación de octava alta y baja.
Programación inteligente de la dirección de las plicas.
Barras divisorias y finales inteligentes.
Número de compás y título de la canción.



PRODUCIDO, MANUFACTURADO Y DESARROLLADO EN CONJUNTO CON SOFT-MUSIC (ALEMANIA)

ES UN PRODUCTO MUSIC-SOFT®

MUSIC-SOFT
Magallanes, 27 - 28015 MADRID


SOLO NUESTROS
"QL"
SE EXPLICAN EN ESPAÑOL
DOCUMENTACION EN ESPAÑOL, CONTENIENDO:
• INTRODUCCION • GUIA DEL PRINCIPIANTE • MANUAL DEL SUPERBASIC
• MANUAL TECNICO DE REFERENCIA
• APLICACION "QL" QUILL • APLICACION "QL" ABACUS • APLICACION "QL" ARCHIVE
• APLICACION "QL" EASEL
MICRO MW WORLD
Modesto Lafuente, 63
Telf. 253 94 54
28003 MADRID
Colombia, 39-41
Telf. 458 61 71
28016 MADRID
José Ortega y Gasset, 21
Telf. 411 28 50
28006 MADRID
Fuencarral, 100
Telf. 221 23 62
28004 MADRID
Padre Damián, 18
Telf. 259 86 13
28036 MADRID
Avda. Gaudí, 15
Telf. 256 19 14
08015 BARCELONA
Ezequiel González, 28
Telf. 43 68 65
40002 SEGOVIA
Stuart, 7
Telf. 891 70 36
ARANJUEZ (Madrid)